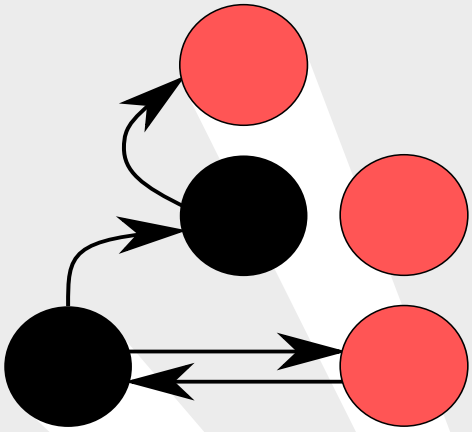
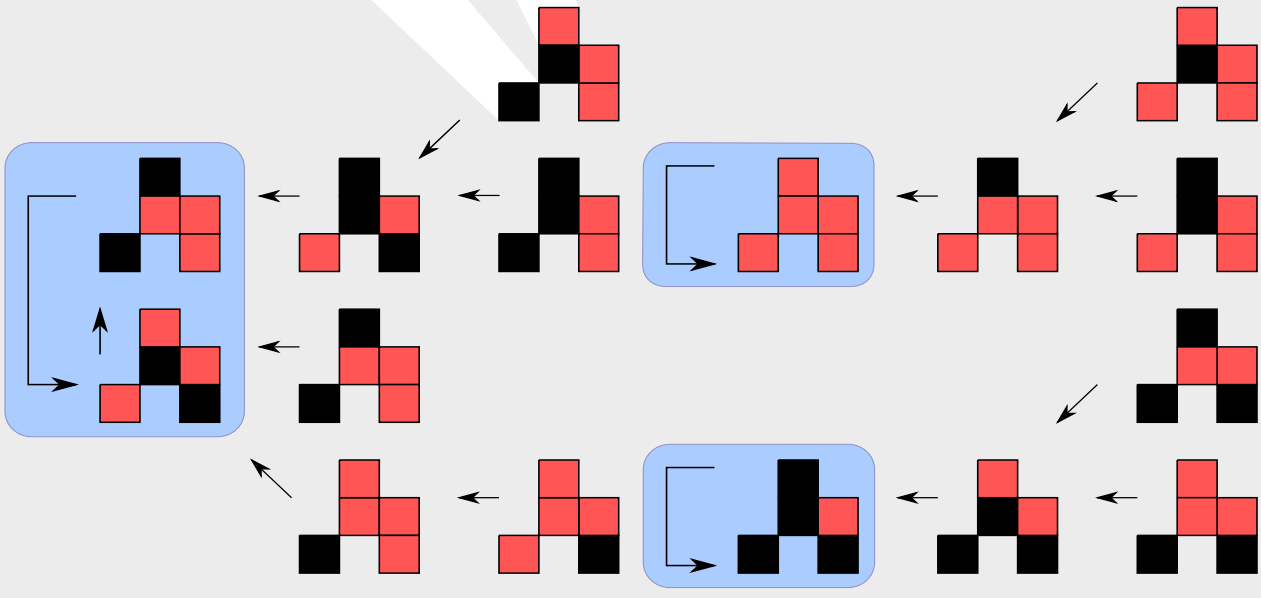
Dynamics of Boolean Networks

Dynamik Boolescher Netzwerke
For graduation as *Doktor der Naturwissenschaften* (Dr. rer. nat.)
accepted PhD-Thesis by Dipl.-Phys. Florian Greil from Darmstadt
Summer 2009 — Darmstadt — D 17



TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Physik
Institut für Festkörperphysik
Hochschulstr. 6
D-64289 Darmstadt



Dynamics of Boolean Networks
Dynamik Boolescher Netzwerke

accepted PhD-Thesis by Dipl.-Phys. Florian Greil from Darmstadt

1st review: Prof. Barbara Drossel
2nd review: Prof. Gernot Alber

Date of submission: 2009, May 25th
Date of exam: 2009, July 15th

— Darmstadt — D 17

Science is what we understand well enough to explain to a computer. Art is everything else we do.

DONALD KNUTH, computer scientist
Preface to the book "$A = B$", 1996.

Abstract

Networks are used to model systems consisting of many interacting units. The topology of networks has been studied extensively while there are still many open questions concerning the dynamics of and on networks. Boolean networks refer to a class of *dynamics on networks*; it is the simplest possible dynamics which allows for analytical studies and easy computer implementation. Applications of networks in general and Boolean networks in particular can be found in numerous fields, ranging from chemistry, biology, economy, computer sciences, linguistics, to sociology and geology.

A classical Boolean network was introduced by STUART KAUFFMAN as a simple model for gene regulation. The Boolean state of a node is determined by a Boolean function whose arguments are the states of its randomly chosen inputs. The inputs are those nodes from which a edge of the network leads to the node under consideration. Key quantities for the dynamics are the number and size of attractors. An attractor is a recurrent set of a network's states. Although the classical model of random Boolean networks with synchronous updating has been introduced in 1969 it took 30 years for an analytical understanding of its key quantities.

The present work studies variations of the classical implementation of a random Boolean network, in particular models with differently defined dynamics (non-synchronous updating, the ensemble of threshold functions instead of all Boolean functions) and systems with other connection patterns (scale-free in-degree distribution). To study the characteristics of the dynamics of those variations both, computer simulations and analytic methods, are used.

For the classical critical Boolean network (with synchronous updating) it is known that both, the mean number and the length of attractors, diverge faster than any power law with the number of nodes. It is shown how this changes for asynchronous updating schemes, in particular for a deterministic asynchronous one. Boolean threshold functions lead new phases of the dynamics which are not predicted by the usual mean-field considerations, real genetic networks might therefore also have a richer dynamical behaviour than the classical dynamical phases. For the scale-free topology, the number of the non-frozen nodes scales in a different way as in networks with a fixed number of inputs. Here, it is possible to analytically explain numerical findings in literature.

Zusammenfassung

Netzwerke werden als Modell für Systeme mit vielen wechselwirkenden Einheiten benutzt. Die Topologie von Netzwerken sind intensiv untersucht worden, wogegen es viele offene Fragen bei der Dynamik von und auf Netzwerken gibt. Boolesche Netzwerke stellen ein Klasse der *Dynamik auf Netzwerken* dar. Eine solche Dynamik ist die einfachste denkbare Dynamik, es sind analytische Untersuchungen möglich und die Umsetzung in Computer-Programmen ist leicht. Anwendungen von Netzwerken im Allgemeinen und von Booleschen Netzwerken im Speziellen finden sich in den unterschiedlichsten Gebieten, angefangen von Chemie, Biologie, Wirtschaft, Informatik, Sprachwissenschaften, bis hin zur Anwendung in der Soziologie und Geologie.

Ein klassisches Boolesches Netzwerk wurde von STUART KAUFFMAN als Modell für die Genregulation eingeführt. Der Boolesche Wert eines Knoten ist durch eine Boolesche Funktion bestimmt, deren Argumente die Werte der zufällig verknüpften Eingänge sind. Eingänge sind hierbei jene Knoten, von denen eine Verknüpfung des Netzwerks zu dem gerade betrachten Knoten führt. Schlüsselgrößen für die Dynamik bilden die Zahl und die Größe der Attraktoren. Ein Attraktor ist eine Menge wiederkehrender Netzwerkzustände. Obwohl das klassische Modell zufälliger Boolescher Netzwerke 1969 eingeführt wurde, dauerte es 30 Jahre bis die Schlüsselgrößen analytisch verstanden waren.

Die vorliegende Arbeit untersucht Variationen der klassischen Implementierung eines zufälligen Booleschen Netzes, insbesondere Modelle mit abweichend definierter Dynamik (nichtsynchrone Aktualisierung, Ensemble der Schwellenwert-Funktionen anstelle desjenigen mit allen Booleschen Funktionen) und Systeme mit anderen Verknüpfungsmustern (skalenfreie Eingangsgrad-Verteilung). Um die Eigenschaften der Dynamik dieser Variationen zu studieren werden sowohl Computer-Simulationen als auch analytische Methoden benutzt.

Es ist bekannt, dass für klassische, kritische Boolesche Netzwerke (mit synchroner Aktualisierung) sowohl die mittlere Anzahl als auch die mittlere Länge der Attraktoren schneller als jedes Potenz-Gesetz mit der Zahl der Knoten anwächst. Es wird gezeigt, wie sich das für andere Aktualisierungsschemata ändert, insbesondere wird dabei ein deterministisch-asynchrones Schema betrachtet. Boolesche Schwellenwert-Netze können zu neuen Phasen der Dynamik führen, die von den üblichen Meanfield-Überlegungen nicht vorhergesagt werden; auch echte genetische Netzwerke könnten daher ein vielfältigeres Verhalten zeigen als die klassischen Phasen der Dynamik. Für skalenfreie Topologie skaliert die Zahl der nicht-gefrorenen Knoten anders als bei Netzwerken mit festen Zahl der Eingänge. Es ist hier möglich die numerischen Befunde in der Literatur analytisch zu erklären.

Contents

1 Introduction

A *complex system* is a system showing a rich dynamical behavior over multiple scales and non-trivial response to changes. A marvelous example for such a system is a living organism. It is often advantageous to describe a complex systems as a *network* where various components of the system, each one represented as a *node* (or a *vertex*) interact. The interactions are abstracted as *edges* or *links*. In mathematical terms a network is a *graph*, consisting of a set of *vertices* and a set of *edges*.

Network science has its origin in *graph theory*. The birth of graph theory is marked by the solution of the so-called *Königsberg bridge problem*. In 1735 the Swiss mathematician LEONARD EULER answered the question whether there exists a walk across the seven bridges such that it never crosses the same bridge twice. LEONARD EULER replaced each of the four land areas with nodes, regarded the bridges in between as links and he obtained a graph, see Fig. 1.1. His proof is based on the simple observation that nodes with an odd number of links must be either the starting or the end point of the journey.

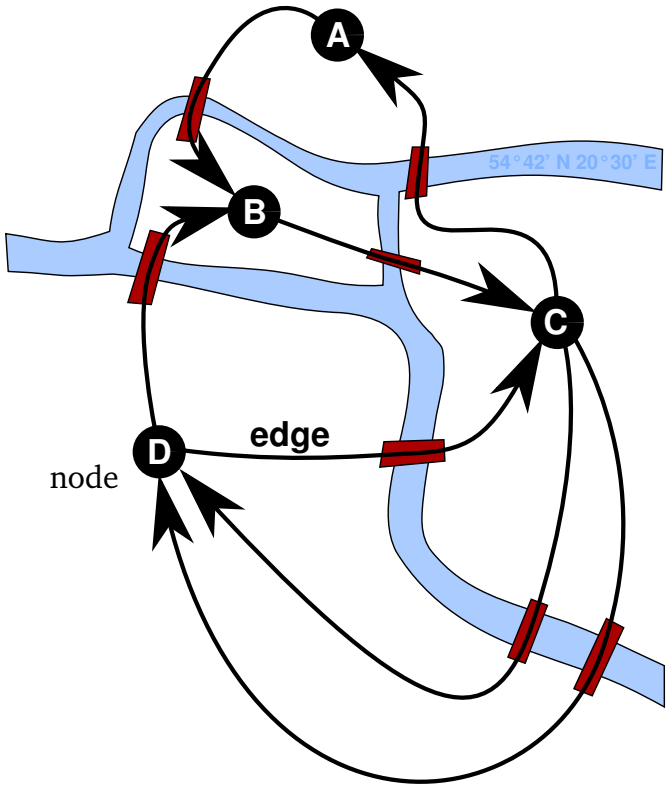


Figure 1.1: Seven bridges in Königsberg as they are arranged nowadays. The Kneiphof island B and the land area C are surrounded by two arms of the river Pregel. Today it is possible to find a route crossing each (red) bridge only once, e.g., $CDCDBCAB$.

After LEONARD EULER, THOMAS P. KIRKMAN and WILLIAM R. HAMILTON studied polyhedra which can be treated as graphs. GUSTAV KIRCHHOFF used graph-theory to calculate the currents in electrical circuits. ARTHUR CAYLEY, GEORGE POLYÁ and others enumerated the isomers of chemical molecules by means of graph theory. All scientists mentioned above considered graphs with a fixed structure. In the 1950's, PAUL ERDŐS and ALFRÉD RÉNYI introduced *random graphs*. They started with N nodes and connected every pair of nodes with probability p, the arising graph will then have approximately $p \cdot N \cdot (N-1)/2$ edges.

Network science is a rapidly evolving field. In the last few years a variety of new developments have been made (for details, see references in Sec. 2). Most of the work provides a framework for dealing with the topological properties of networks. The *topology* is the manner in which the units of the network are arranged.

The next step is to analyse the dynamics on a network. Each node now holds a variable and additionally an interaction rule is defined. An example beside many others are networks formed

by chemical reactions, see Tab. 1.1. If the the system is not too large, one can numerically study such a system by Monte-Carlo simulations, keeping track of position and velocity of each molecule. A reaction takes place when two molecules are sufficiently close to each other. This dynamics can be abstracted to differential equations for the time evolution of the reactants. The concentration is now the variable of each node, the interaction rules are the rate equations for the reactants. Abstracting the underlying mechnisms even further, one ends up at the level of discrete dynamical switches. Analytical work in this field is still in its infancy. The present work aims to contribute to understanding of the dynamics by focusing on the simplest possible dynamics namely a *Boolean* one. A Boolean variable can take one out of two possible values, either true (1) or false (0). The Boolean approach exploits insights from statistical physics in order to uncover how the details of such a toy model influence the main features of the dynamics. The idea is to learn more about the fundamental processes on networks.

The present work is organised as follows: After motivating the network approach by examples from different fields (see Tab. 1.1) in more detail, the most important definitions and concepts are presented in Sec. 2. Both aspects of networks, *topology* and *dynamics*, will be elucidated.

Table 1.1: Various examples of networks. For each system the entity corresponding to a node and the interaction defining an edge is specified. In Sec. 2 more details about those examples are given.

field	nodes	edges
chemistry	reactions	reactants occurring in an reaction
molecular biology	transcription factor	interaction partners
	proteins	reactions leading to a different function
	metabolism substrates	chemical reactions
neurobiology	neurons	synapses and axons connecting them
	groups of neurons	influence between those groups
ecology	populations	predator-prey relationships
economy	companies	trading relations
	airports	direct flights
telecommunication	computers	wires, fibres, radio connections
	web pages	hyperlinks directed to other pages
	mobile phone numbers	calls in between them
electronics	resistors, inductors, capacitors	wires or circuit board paths
linguistics	words	proximity within a text
	words	semantic relation
sociology	authors	being coauthor
	actors	playing in the same movie
	persons	communicative action of any kind
geology	epicentres	earthquakes occurring successively

The main part starts with the classical implementation of a Boolean network and the features of critical Boolean networks (Sec. 3). Then, three different modifications of the classical implementation of a Boolean network will be considered: The updating scheme of the dynamics is varied (Sec. 4), the choice of the internal dynamics is changed (Sec. 5) and finally topological modifications are considered (Sec. 6). The present work concludes with an outlook (Sec. 7).

2 Networks and their dynamics

In this chapter, the *topology* (Sec. 2.2) and *dynamics on networks* (Sec. 2.3) will be introduced. To motivate the use of network models in different fields, the examples from Tab. 1.1 will be explained in more detail (Sec. 2.1), emphasising possible applications of this work.

2.1 Some examples of networks

The terms *node* and *edge* are already defined in Sec. 1 to be the basic ingredients of a network. In the following, *directedness* and *weight of an edge* will be defined, as those two properties vary for different real-world networks.

An edge can be either *undirected* or *directed*, the latter means that an orientation is assigned to each edge in the network. A directed edge $e = (x, y)$ is also called an *arc* pointing from node x to node y, y is said to be a *direct successor* of x or, vice versa, x is the *direct predecessor* of y. Throughout this work, *digraphs* will be considered, i.e., graphs with arcs as edges as this is the more general representation. All undirected networks can be mapped to directed ones by introducing two arcs in both directions. In a *weighted graph* each edge is associated with a label *weight*, which is usually a real number specifying for example the *cost* of an edge.

Networks in chemistry

Reaction schemes with many reactions, intermediate products and possibly catalysts can be described as networks. One formalisation would be to treat each chemical reaction as a node. The reactants occurring in the reaction equation form directed edges. An arrow pointing from reaction R_1 to reaction R_2 means that some product from R_1 serves as an educt in R_2. The concentrations of the chemical species, say in number of molecules, would specify the weight of the edges here.

Instead of assembling an exact function to describe the interdependence of two reactants, it is often sufficient to know which reactants influence each other. Ralf Steuer and coworkers developed a generic method [120, 119] to analyse the dynamics in dependence of reaction parameters. The method is based on studying the Jacobian matrix of the system and its linear stability analysis. They are interested in *bifurcations*, qualitative transitions of the dynamics. The advantage is that only the stationary concentrations and the dependence of the reaction rates on the concentrations are required. In other approaches many more parameters are needed.

Networks in molecular biology

One big goal in molecular biology is to construct a *virtual cell*. Cellular processes form networks on many levels. For a brief introduction into the bio-chemical details of this field see App. A. In principle, genes and all their products interact, one usually distinguishes four levels:

1. A protein is a *transcription factor* if it binds to specific DNA sequences and thereby controls the transcription of genetic information from DNA to RNA (see App. A for the biological details). Thus, a transcription factor can activate or inhibit the formation of other proteins

If a protein is produced, the gene is said to be *expressed*. Proteins are themselves products of genes, thus genes regulate each other's expression. This interwoven relationship is called a *gene regulatory network*: The nodes are the transcription factors and the genes with whom a transcription factor can interact are connected by the edges of the network. This is the level of the *genome*, which refers to the full set of genes of an organism.

2. Beside that, the intermediate products ("transcripts") occurring during the assembly of proteins may also influence each other. This is the level of the *transcriptome* which is the set of all transcripts. At this level, the so-called *RNA interference* occurs, small pieces of RNA (the transcripts) take part in gene regulation.[1] This is important as the defending mechanisms against parasitic genes e.g. from viruses is based on RNA interference.

3. Proteins can be changed even after they have been put together by the cell's machinery. Proteins may modify other proteins leading to a modified function. Alternatively, multiple proteins may attach to each other, building up a *protein complex*. All those so-called *post-translational interactions* form edges in the *protein-protein interaction network* or just *proteome*. There are around 10^3 to 10^4 active proteins in a cell and therefore this many nodes in the network.

4. The *metabolism network* or *metabolome* is composed of chemical processes by which energy is stored or released within a cell. The ubiquitous substrates (like ATP, ADP or NADP) are the nodes, the edges represent the (often directed) chemical reactions in which these substrates participate. The metabolome incorporates of the order of 10^3 metabolites per cell. A sequence of enzyme reactions is called *metabolic pathway*. Such pathways are vital for the development of drugs against illnesses. In a network with metabolites as nodes and reactions as edges such a pathway may be understood and possible drug targets can be predicted.

Networks in neurobiology

A widely used term for either a biological or an artificial system in neurobiology is a *neural network*. Nodes can be either single neurons or groups of nearby (and possibly functionally associated) neurons. The connections between those units form the directed edges.

Biologically, neurons are nerve cells processing and transmitting information by electro-chemical signalling. Connections between two neurons are mediated by *synapses*. The weight of the edges may be changed with time, this is referred to as *learning*. For example, in [117] both the structural properties of both anatomical and functional brain networks are reviewed for intermediate sized systems (the nodes are areas of neurons) consisting of approx. 10^3 vertices and 10^4 edges.

The first artificial neuron as a mathematical model was introduced in 1943 by Warren McCulloch and Walter Pitts as a unit with many inputs and one output [89]. Traditionally, nodes in artificial networks can be sorted into *neural layers*. Only nodes between subsequent layers are connected by edges. Furthermore, one usually distinguishes two modes of operation: First, one trains such an artificial network to adjust the internal functions of the nodes to a desired output. Certain input-patterns from the training set are concatenated with an known-output pattern. Then, in the application mode, the functions are used to determine the output pattern,

[1] More specifically, RNA interference is an RNA-dependent gene silencing process in eukaryotic cells. It is controlled by the RNA-induced silencing complex (RISC) and is initiated by short double-stranded RNA molecules.

irrespective whether the input configuration was part of the training set or not. Artificial neural networks are applied as non-linear statistical data modelling tools. Applications range from pattern and object recognition in computer vision (for robots or autonomous vehicles) over trajectory or time series prediction, speech syntheses or recognition to various optimisation tasks in automatic control engineering.

Networks in ecology

In an habitat like a lake or an island, multiple species interact in a *food web*. This network quantifies the predator-prey relationship. The species are abstracted as nodes, the directed edges represent the nutrient flow between them. Primary producers supply complex organic substances by incorporating inorganic matter and energy. These organisms typically are photosynthetic plants. All other organisms obtain their energy by consuming either only plants (*herbivores*) or other living animals (*carnivores*). Beside those, the *detritivores* consume dead biomass.

In the traditional view of ecologists it was observed that the more complex the ecosystem is, the more stable it seemed to be against population fluctuations of a species. However, randomly linked nodes with the standard Lotka-Volterra population dynamics showed the opposite.[2] The relationship between stability and complexity is a challenging and important issue in ecology, for a current study see [122] and references therein.

Networks in economy

Nodes can be companies where an edge would mean that two connected companies share for instance a member of the management (undirected edge) or that one company buys products from the other one (directed edge). Similarly, an *economic branch network* can be constructed with nodes being trades and edges representing companies trading on the same market [38].

The International Air Transport Association (IATA) maintains a database which contains a list of all possible direct flights between two airports. Vittoria Colizza and coworkers studied epidemic processes on that network in order to unveil the role of the network structure on the spatio-temporal pattern of emerging diseases [41]. Furthermore, they evaluated the reliability of epidemic forecasts and their outbreak scenarios.

Telecommunication networks

One example for a telecommunication network is the *World Wide Web* (WWW) or subsets thereof like single websites. The nodes of the network are the web pages and the edges are the hyperlinks within each page pointing from one document to another. Studying these networks prospers as they are easy to probe. Therefore it is also the largest available network—one should be aware that even powerful search engines can index only a small percentage of the WWW.

In contrast to the WWW, the *Internet* is a network of physical links between the client computers, servers, routers and other devices. Those are the nodes, the edges are the cables and optical fibres to connect them.

A variation without wires are so-called *ad-hoc networks* which recently gained a lot of attention in computer science. An ad-hoc network is a decentralised network in which each computer is willing to forward data to other computers over a radio communication connection. The con-

[2] The coupled non-linear differential equations $\dot{x} = (\alpha - \beta y) \wedge \dot{y} = -y(\gamma - \delta x)$ are commonly referred to as *Lotka-Voltera equations*. They were proposed independently by Alfred J. Lotka and Vito Volterra in the 1920's. One can interpret y as the number of individuals of some predator and x as the number of its prey.

nections between nodes are thus determined by the availability of computers nearby. Telecommunication carries are interested for instance in the spreading of computer worms or viruses in such systems.[3] The predicted initial growth of the epidemic was found to be significantly slower than the exponential growth observed in the Internet [98].

Similarly, a large network can be constructed from telephone call patterns, where nodes are phone numbers and every completed phone call is an edge, directed from the caller to the receiver [80, 99].

Electrical networks

Electrical elements such as resistors, inductors and capacitors can be the nodes of an *electrical network*. On micro-controllers such networks can be really large (10^8 and more components per chip) and have therefore been subject to studies dealing with their topology, e.g. [129].

A *power grid* is the network for delivering electricity from suppliers to consumers. Nodes can be power plants, homes, businesses, subnetworks and are connected by cables of different kind. A classic distribution grid is a *tree*, in such a network there exists exactly one way from the power plant to all consumers, see Sec. 2.2 for a strict definition. Recent large-scale power outages triggered studies on the structural vulnerabilities of power grids [4] where redundancy is required most urgently to prevent blackouts.

Networks in linguistics

The human languages offer several possibilities for networks. There are studies taking words as nodes and edges occur if two words appear within a certain proximity in a sentence. Alternatively, free semantic associations can be used to defined the links. An example of a semantic network is *WordNet*, a lexical database of the English language [127]. Words are grouped into sets of synonyms which are the nodes of the network. The various semantic relations between these synonyms form the edges, examples are termed *meronymy* (A is part of B), *hyponymy* (A is a subtopic of B) *synonymy* (A is the same as B), and *antonymy* (A is opposite of B).

Networks in sociology

Even social systems can be regarded as networks. The individuals are the nodes and the edges represent some sort of social interaction. Researchers in this field also can take advantage of large databases, such as:

- Databases like *Citebase* [40] harvest pre- and post-prints from various archives for different fields. For instance, nodes stand for published articles and a directed edge represents a reference to a previously published article. Alternatively, one can also extract co-authorship graphs of scientists writing papers together being connected by an undirected edge.

- The *International Movie Database* [65] is analysed in many studies. An actor collaboration network can be reconstructed, in which actors casted jointly are assumed to know each other, see the Sec. 6.1 for some details.

- Then, there are quite a number of dating sites analysed as obtained from Internet portals. For example, PETTER HOLME and coworkers looked at various features of social networks.

[3] A *computer worm* on the one hand is a self-replicating self-contained computer program using the network to send copies of itself to other computers. On the other hand, a *computer virus* spreads by inserting copies of itself into other executable code or documents on both the network and a single computers.

For example the *reciprocity of communicative action* measures the direction of the communication flow between any two users. One usually assumes acquaintance networks to have a high level of reciprocity, however, interaction through such a website seems to put less social pressure onto each person to respond to a communicative act than in a face-to-face or telephone encounter [62].

- Another interesting phenomena is the occurrence of rhythms of social interactions. The group around BERNADO HUBERMAN found a clear weekend pattern while studying messaging within a massive online friendship site providing insight to the social life of the users [56].

There are also sociological studies with network models carried out using questionnaires and personal interviews: In [85], a data set of $\sim 10^4$ Swedish citizens and their sexual contacts was analysed over a time interval of 12 months, finding that the topology of this network varies from the one of friendship networks. A undirected link arises as the relation is initiated and disappears when it is terminated. Some severe diseases like AIDS spread on a network of sexual relationships. To tackle this it is essential to understand the underlying network's structure .

Networks in geology

Seismic data can be mapped onto a directed network. SUMIYOSHI ABE and NORIKAZU SUZUKI introduced the following method [41]: A region is divided into cubic cells, each cell is regarded as a node if earthquakes occurred therein. If two successive earthquakes occur in two cells, the corresponding nodes are linked by an directed edge representing an event-event correlation. Their approach offers a criterion to compare real datasets with an earthquake model on a lattice, as introduced by TIAGO DE PAULA PEIXOTO [103].

Networks constructed from general time series

There has been recent work on general time series constructing complex network from the data. One approach is the so-called *visibility graph* network [82]. Every node corresponds to a data point at a given instance of time. The values are treated literally as a landscape. All the other values an observer sitting on a given value can see are connected. That is, two data points are visible to one another in case all ordinates of the points in between are smaller than the straight line connecting the first two points. The aim is to be able to characterise the series using methods from graph theory [128].

2.2 Quantifying the Topology

The first approach to a yet unknown network is to merely describe its geometry. Some common topological terms of graph theory will be defined in this section. All of the previously mentioned empirical networks have been classified according to its topology, the review [6] gives an concise overview over some of the results.

A particular network is completely described by its *adjacency matrix* $A = (a_{ij})_{N\times N}$ if N is the overall number of nodes present in the network. Each element a_{ij} of the matrix A equals the number of edges connecting vertices i and j. An entry can also be seen as the *weight of an edge* if it is a non-integer number. Except in Sec. 5 the present work focuses on networks without weight, but multiple links between the nodes are always allowed.

The adjacency matrix of an undirected network is symmetric. The directionality of the edges depends on which kind of empirical network one considers: For example, the WWW is directed as source and destination pages are well-defined. Conversely, social networks are usually non-directed, since the relationships there are assumed to be more mutual. The main part of the present work will always deal with directed networks as undirected networks can be mapped onto directed ones.

If the adjacency matrix of an unweighted network is sparse, one usually writes for each node i just a list of the non-empty columns of the matrix for each row i. The present work will always use this representation as it is computationally efficient.

Local characteristics of a vertex

The simplest property of a node i is its *degree* k_i, the total number of the edges attached to the node. In a *directed network* (also referred to as *digraph*) there are two kinds of degrees for each node i, namely the number of incoming and outgoing edges, k_i^{in} and k_i^{out}, respectively. The histogram of degrees accounts for how many nodes of a certain degree exist. In the directed case the *in-distribution* $P_{\text{in}}(k)$ and the *out-distribution* $P_{\text{out}}(k)$ have to be distinguished. Classical random networks have Poissonian degree distributions, however, in Sec. 6 a scale-free distribution, $P_{\text{in}}(k) \propto k^{-\beta}$, will be considered.

Nodes with extreme degrees have special names: A node with outgoing edges only is a *source node*, in real genetic networks for instance a source node would represent an external influence. The opposite is a *sink node* which only has incoming edges. A *hub* is a node with an extremely high degree, hubs are usually least abundant. The degree-threshold above which a node is defined to be a hub varies from problem to problem.

A *clique* in an undirected network is a complete subgraph, i.e., every pair of nodes is connected. Let κ_i be the number of edges between the nearest neighbours of node i and the node itself. The *local clustering coefficient* C_i quantifies how close the node and its neighbours are to being a *clique*. If k_i again is the degree of node i, the local clustering coefficient is given by

$$C_i \;=\; \frac{2\kappa_i}{k_i(k_i-1)}. \tag{2.1}$$

This property can be made global by averaging over all nodes, $C = N^{-1} \cdot \sum_{i=1}^{N} C_i$.

In 1998, DUNCAN WATTS and STEVEN STROGATZ invented this a network model with adjustable C, a so-called *small-world network* [123]. In their work, they started with n nodes arranged in a ring an being connected with its k neighbours, see Fig. 2.1. They interpolated from this regular ring to a random network by rewiring all links per node to a random node on the ring with probability p. For intermediate values of p a highly clustered graph with a small characteristic *path length* is obtained, the details will be subject of the next section. In fact, many real graphs exhibit the *small-world property*, namely both small characteristic path length and a high average clustering coefficient.

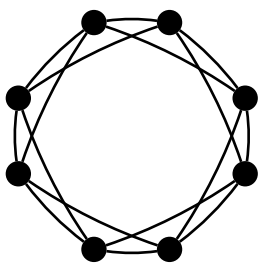


Figure 2.1: DUNCAN WATTS's and STEVEN STROGATZ's small-world network with $n = 8$ and $k = 4$.

Properties connected with paths

A *path* is a sequence of nodes such that there is a link directing from a node to the next one in the sequence, illustrated in Fig. 1.1. Many of the properties of path are important for analysing the dynamics of networks in *state space*.

When starting the sequence at an outgoing link from a given node $\star$ and following such a path, it might happen that there are only paths to a subset of nodes reachable from node $\star$. A *connected network* is a graph where a path exists between every pair of nodes. If the network is directed and the same property holds, the network is said to be *strongly connected* as the condition is harder to fulfil.

The *average path length* $\langle l \rangle$ is the mean over all shortest paths between any pairs of nodes and is thus a global property of a network. For small-world networks with any kind of degree distribution the average path length scales as $\langle l \rangle \sim \ln(N)$ [123].

Another feature incorporating paths is the *betweenness centrality* which measures the influence of a given node on the overall connectivity of a network. The betweenness centrality B_i of node i is defined as

$$B_i := \frac{\text{number of all shortest paths from node } a \text{ to } b \text{ passing through } i}{\text{overall number of shorted paths between } a \text{ and } b} \tag{2.2}$$

Calculating the betweenness centrality is numerically expensive as it involves calculating the shortest paths between all pairs of nodes. The standard method for directed networks is the *Floyd-Warshall algorithm*[4]. It takes of the order of $\mathcal{O}(N^3)$ steps which is remarkable few as there can be up to N^2 edges.

Furthermore, the betweenness gives a hint about the information flow through the nodes when stepping forward to a dynamic on a network, which is done in Sec. 2.3. It is also of practical relevance: Petter Holme and coworkers showed that the most ubiquitous substrates in bio-chemical pathways often have the highest betweenness [63].

Network building blocks

Subgraphs occurring statistically more frequently than they would in a degree-preserving randomised version of a given network are called *motifs* [94, 95, 112]. Analogously *anti-motifs* are defined, those are subgraphs occurring less often than one would expect in the random case. Each type of network seems to display its own set of characteristic motifs. Fig. 2.2 summarises abundant motifs as studied by Uri Alon [12]. The idea is that motifs might be seen as functionally separable subgraphs that regulate different processes and thus act as functional building blocks.

- The simplest motif incorporating only one node is *auto-regulation*, see Fig. 2.2(1).
- The *feed-forward loop* (FFL) is also important both for real networks and for the mathematical models, compare Sec. 2.3. It can act as a delay filter, in the sense that the left-most

[4] One defines $(a \xrightarrow{x} b)$ to be the shortest path from node a to b using only nodes with identifiers $\{1 \dots x\}$ as intermediate points. The algorithm can now be summarised (for unweighted graphs) in the following recursive formula:

$$i \xrightarrow{k} j = \min\left(i \xrightarrow{k-1} j, -\left(i \xrightarrow{k-1} k \; + \; k \xrightarrow{k+1} j\right)\right)$$

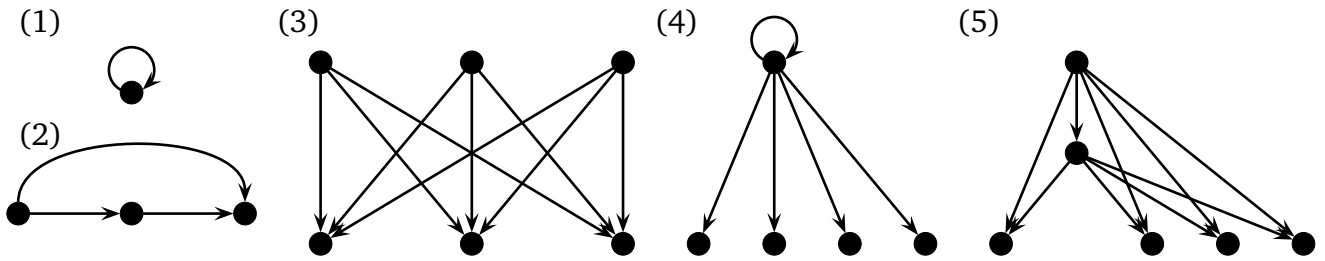


Figure 2.2: Some abundant motifs. The arrows denote a positive influence. Auto-regulation (1), feed-forward loop (2), dense overlapping regulon (3), single-input module (4) and a multi-output feed-forward loop (5).

node in Fig. 2.2(2) has to deliver a signal for a time interval long enough such that the information of the current state has time to propagate through the middle node.

- A *dense overlapping regulon* (DOR) can be found in neural networks and consists of layers, see Fig. 2.2(3). From the theoretical point of view, such a motif is formed by interwoven trees, in the example the three nodes in the upper layer are the *root nodes* of the trees.
- In a genetic network, a *single-input module* (SIM) can control the temporal order of gene activity. In Fig. 2.2(4) there are five target nodes regulated by one auto-regulated master node. If one imagines the downward pointing arrows to have a time delay proportional to the length of the edges, the node right below the master node receives information first while the rightmost node is the last one to be regulated.
- The last example for a network motif to be introduced here is the *multi-output feed-forward loop*. One recognises the coherent FFL within, consisting of the three nodes on the left in Fig. 2.2(5).

The greater amount of the nodes that are involved in a motif, the higher the probability there will be to count patterns multiple times. In the multi-output feed-forward loop the statistics already contains the FFL. This raises the question whether a degree-preserving randomisation really is a good basis for comparison which subgraphs are defined to be motifs. There is also some evidence that motifs in real networks may result of the evolution of networks and might only weakly depend on certain function [39].

The concept of finding building blocks is also interesting from the mathematical point of view. In Sec. 3 so-called *relevant components* will be introduced which have much in common with the motifs occurring in real-world networks, in particular feedback-loops.

Zooming out – an even more coarse grained view

A *community* is a group of nodes which are more densely connected among themselves than to other nodes. Depending on the method used, communities can be overlapping on disjunct. In this context, *modularity* is often used as a benefit function measuring the quality of a division into communities. Positive values of the modularity mean that the number of edges within the group of nodes is higher than the number of edges in a degree-preserving randomisation of the subgraph. A possible way of defining the modularity Q of a division is to calculate the fraction of the edges within a given group minus the expected fraction of a random division,

$$Q = \frac{1}{2 \cdot L} \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2 \cdot L} \right) \delta(c_i, c_j). \tag{2.3}$$

In this formula, the sum runs over all pairs of nodes, (ij). Overall, there are L links. Each node i is tagged by c_i to belong to a certain group. The Kronecker delta $\delta(c_i, c_j)$ delivers a contribution to the sum if two nodes i and j belong to the same community.

Especially in the case of a directed network which is not strongly connected it is possible to break down the network into several *partitions* according to how their nodes are reachable by paths from other nodes. Albert-László Barabási [18] coined the term *continents* for the partitions.

The network may have one or multiple subsets of nodes which are indeed strongly-connected components, those components all belong to the network's *central core*. Each strongly-connected component furthermore may be attached to an *in-component* containing nodes with paths leading from them into the strongly-connected component, but not vice versa. The aggregation of all in-components forms the *in-continent*. Nodes in the in-continent are arranged such that following the links eventually leads into the central core, but starting from the central core does not allow mean that one can return to the in-continent by following the links. All the terms of this paragraph prove useful when considering state space networks [27].

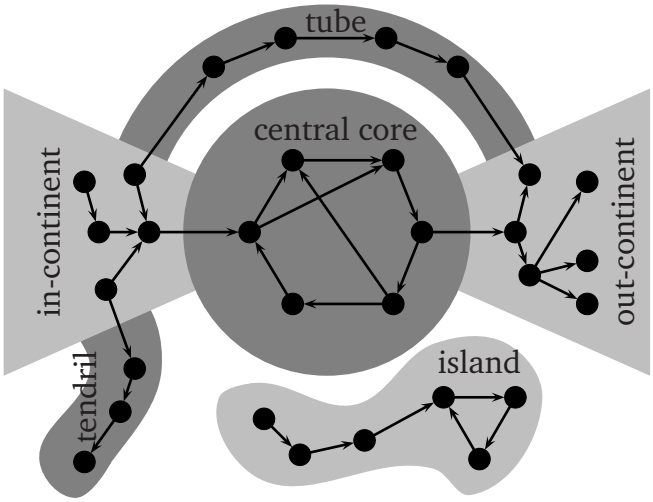


Figure 2.3: Different network continents for a directed network, freely adapted from a sketch in [18].

In contrast, all nodes of an *out-component* can be reached from the attached strongly connected component, but it is not possible to turn back into after leaving it. The set of all out-components is the *out-continent*. Some additional arrangements can occur: *Tubes* are connections between the in- and the out-continent, *tendrils* of paths may lead out the in-continent but not into the central core (or paths leading into the out-continent but not coming from the core). Finally, disconnected *islands* of nodes can occur. Those nodes are not strongly connected and thus have not been captured in the above classification.

2.3 Introducing dynamics

The term *dynamics on networks* refers to different types of processes that take place on networks of a given topology. In contrast to that, the term *dynamics of networks* describes changes of the topology with time. The present work will focus on the first issue.

To construct a dynamics on a graph, an additional variable σ_i, the *value of a node*, has to be introduced. Furthermore, an interaction rule $f_i(\sigma_1, \ldots, \sigma_N)$ has to be defined, the interaction rule can be the same for all nodes or the rule can vary according to some distribution where different nodes may have different functions. There are various possible interaction rules, five common models will be briefly recapitulated here.

Oscillator networks

A way of defining dynamics on a network is to implement oscillators at each node as dynamical elements. The coupling between the oscillators is provided by the links and may include time delays. The network can be either homogeneous (with functionally identical oscillators at each node) or inhomogeneous where some properties of the oscillators vary between nodes. Furthermore, the topology of the network can be defined in various ways, see Sec. 2.2. A feature to be analysed is whether the constructed network synchronizes.

Reaction-diffusion models

As it is computationally expensive to keep track of position and velocity of each molecule, *reaction-diffusion models* are a common approach to study bio-chemical interactions. Such models can be implemented on networks, mostly on undirected graphs. At the nodes, reactions between substances take place. The diffusion spreads the substances along the edges. The additional variable $\sigma_i(t)$ is the (continuous) concentration of a (chemical or biological) species of node i at time t. The interaction rule is usually assumed to be the same at all nodes, as chemical relations do no depend on the position in space.

A continuous implementation a reaction-diffusion model includes many details and yields quantitative predictions, but experimental data that provide parameter estimates are scarce. The number of parameters such as production and decay rates is a challenge for both experimentalists who have to measure them and theorists who have to guess realistic values for the missing parameters [119, 120].

Dynamics of walkers on networks

Agent based modelling deals with spatially distributed agents such as animals, peoples or companies. When the motion of the individuals is an important factor, one uses *walker models*, where particles move on a graph. There might be only one or multiple species of walkers, they may exclude each other or react with each other or play games against each other. Compared to reaction-diffusion models, the interaction rules are defined in a different way: agents may disappear or have internal states which is not true for reactants.

Agents located at nodes

Another agent-based dynamics can be defined by sequential site-exchange rule. The agents are assumed to be fixed in space at a certain node of the network. The individuality of the agent is represented by the state σ_i of node i where the agent is located, the interaction between the agents is described by an transfer diagram. One implementation of such an agent-based system is used for *epidemic models*, a simplified picture of describing the transmission of diseases through individuals.

A widely used implementation is the so-called *SIR-model* as proposed by William Ogilvy Kermack and Anderson Gray McKendrick [73]. Each agent can be part of one out of three disjoint compartments:

1. A *susceptible* individual, $\sigma_i = S$, can contract the disease and become infected.
2. An *infective* individual, $\sigma_i = I$, is capable of transmitting the disease.
3. A *removed* individual, $\sigma_i = R$ already had had the disease and is either dead ("removed") or has recovered or became permanently immune ("resistant").

Table 2.1: Truth table of network shown in Fig. 2.4. For all five nodes the value σ' at time $t+1$ (dark gray columns) is given as a function of the values of the $K = 2$ input nodes at time t. For pedagogical reasons the functions are chosen to depend on less than K inputs, the important ones are shaded in light gray. If a node α depends on nodes β and γ, then it is $\sigma'_\alpha = f(\sigma_\beta, \sigma_\gamma)$.

σ_a	σ_b	σ'_a	σ_a	σ_c	σ'_b	σ_c	σ_d	σ'_c	σ_c	σ_e	σ'_d	σ_d	σ_e	σ'_e
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	1	1	0	1	1	0	1	0	0	1	1
1	0	0	1	0	0	1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Other variations of epidemic models assume additional stages (like passively immune infants M or exposed individuals E) and/or different transition schemes, the most common ones are SIS, $SIRS$, $SEIS$, $SEIR$, $MSIR$, $MSEIR$ and $MSEIRS$.

Boolean dynamics

The physicist's approach is to start with the simplest model possible and to see which ingredients of the model are necessary to reproduce some given behaviour. The simplest possible choice for any dynamics is to have exactly two states, $\sigma_i(t) \in \{0, 1\}$, each node can be either "true" (1) or "false" (0). Variables with only two possible values are called *Boolean*, the corresponding logical calculus of truth values was developed by George Boole in the 1830s [31]. In the present work the focus lays on directed Boolean networks with non-homogeneous interaction rules, i.e., each node i has its own *interaction rule* f_i. The present work is based on a Boolean network widely named *N-K-model*.

2.3.1 The N-K model

Stuart Kauffman constructed the *N-K-model* in 1969 [67] as a simple model for gene regulation.

He assumed that each gene can be represented by a Boolean node. An "on"-node, $\sigma_i = 1$, would correspond to a gene that is transcribed (also called an *expressed gene*) while an "off"-node, $\sigma_i = 0$, represents to a gene that does not undergo transcription (in App. A the molecular basis of gene regulation is concisely presented). However, it is known nowadays that genes can show nearly continuous levels of expression. This finding does not diminish the fascination of Boolean networks as they form a very general model applicable to a wide range of systems.

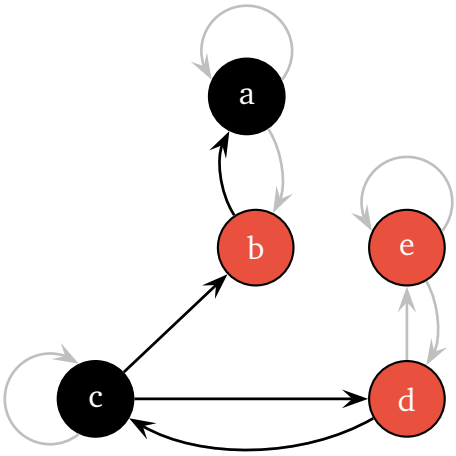


Figure 2.4: Example of a Boolean network consisting of $N = 5$ nodes. Each node i has a Boolean value $\sigma_i \in \{0 = \text{black}, 1 = \text{red}\}$ and a function f_i specified in Tab. 2.1.

Mathematically speaking, a *Kauffman network* ($\equiv N$-K-model) is a directed graph with N Boolean nodes and $(N \cdot K)$ random links in between. There are three key ingredients:

- The in-degree for all nodes is fixed to be K inputs per node, $K = 2$ in Fig. 2.4.

- A a Boolean value, $\sigma_i \in \{0,1\}$, is assigned to each node.
- A Boolean function f_i determines the value of node i at the next time step and can be specified by a truth table like Tab. 2.1.

The second restriction in connection with the last one allows for effectively less than K incoming links as the function f_i can be chosen such that a node does react not at all to changes of a certain input. Fig. 2.4 shows a pedagogical example with $K = 2$. For simplicity, the functions are chosen such that some links (those in gray in Fig. 2.4) do not contribute to the dynamics. In the example, the remaining links just copy the value of the input nodes.

Assuming that $d_{i,1} \dots d_{i,K}$ are the indices of the nodes serving as *inputs* of node i, the time evolution of the value σ_i can be formalised as

$$\sigma_i(t+1) := f_i\left(\sigma_{d_{i,1}}(t), \sigma_{d_{i,2}}(t), \dots, \sigma_{d_{i,K}}(t)\right). \tag{2.4}$$

This defines a *discrete dynamics* which is usually evaluated synchronously. However, in Sec. 4 the choice of other updating schemes will be considered.

Trajectories in state space

The main interest of the present work is the dynamics of Boolean networks. A *state* of a Boolean network is the vector

$$\boldsymbol{\sigma}(t) \;:=\; \{\sigma_1(t), \sigma_2(t), \dots, \sigma_N(t)\} \tag{2.5}$$

of the values of all N nodes at a given time t. The set of all 2^N possible states forms the *state space* of the network. The time evolution of the system is represented by a *trajectory* in state space.

The state space forms another network, the *state space network*, which has to be strictly distinguished from the physical network representation. Figure 2.5 shows the state space network of the example with $N = 5$ shown in Fig. 2.4, due to symmetry $16 < 2^5$ states are sufficient.

As the state space is finite and the dynamics deterministic (see Eq. (2.4)), eventually (a series of repeating states can be observed, an *attractor*. The general definition for dynamical systems says that an attractor is a set of states to which the system evolves after a long enough time. Furthermore, this set is attractive in the sense, that there are also states leading to the attractor without actually being part of it. Those *transient states* lie in the *in-continent* of the attractor. The set of all transient states is usually called the *basin of attraction* of a given attractor. Repeating states without any basin of attraction are called *cycle*. However, in some publications, the term attractor is also used for cycles, the contraction of state space to an attractor is disregarded then.

Assuming the attractor has length A and the transient time is called T, then the condition for the attractor states writes

$$\boldsymbol{\sigma}(t+A) = \boldsymbol{\sigma}(t) \qquad \text{for } t > T. \tag{2.6}$$

Classifying the dynamics

It is possible to distinguish two phases of the dynamics in random Boolean networks with the help of the (normalised) Hamming distance (for a review see [10]). Two copies of a single network realization are prepared from a given ensemble in different initial states, $\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}$.

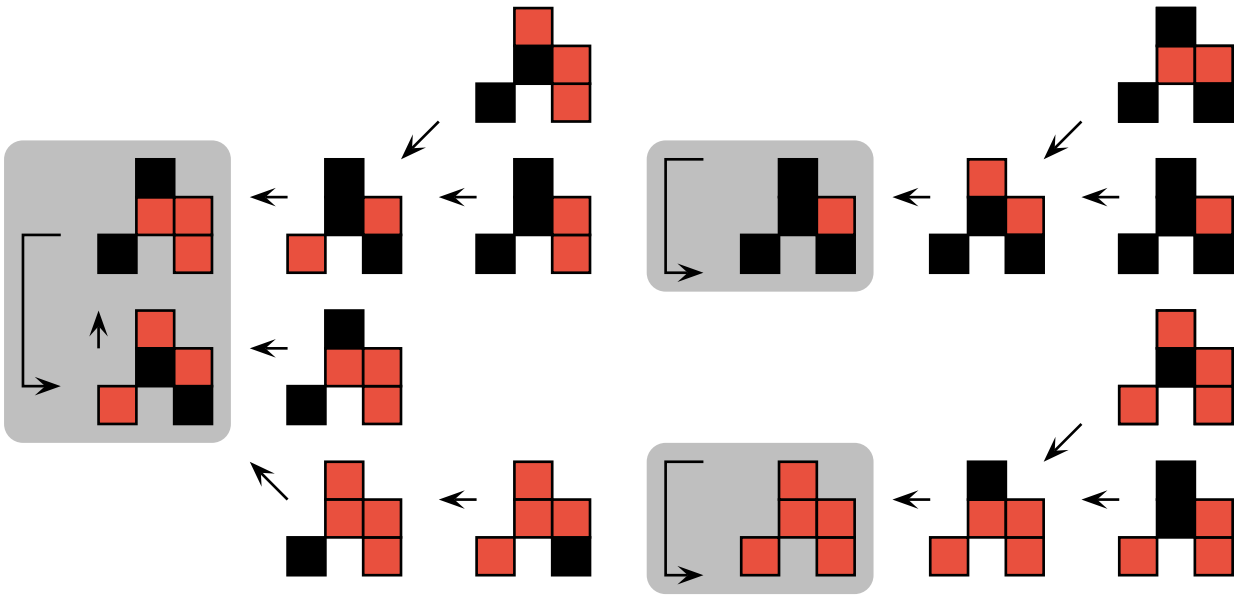


Figure 2.5: The sketch shows the state space for the network from Fig. 2.4 under synchronous updating. There are three attractors (gray shaded areas): A 2-cycle and two fixed points. Each state is a snapshot of the system in Fig. 2.4.

The normalised Hamming distance h_t between the two copies is the fraction of nodes having a different node value:

$$h_t = N^{-1} \sum_{i=1}^{N} \left(\sigma_i^{(1)} - \sigma_i^{(2)} \right)^2 . \tag{2.7}$$

For small h_t, the probability that more than one input of a node differs in the two copies can be neglected and the time evolution can be written as $h_{t+1} = \lambda h_t$ with the *sensitivity* λ [113].

Small sensitivities, $\lambda < 1$ are typical for the *frozen phase*. In this case a perturbation of a node's value propagates to less than one other node on average per time step. Thus, in a large network, the vast majority of nodes will become frozen on the long run. A node is "frozen" on an attractor if it stops changing its value after some *transient time*. If same realization is started in two different initial states, the states of all nodes apart from a finite number become

Table 2.2: Historic review of the results for attractor properties, the mean number $\langle v \rangle$ and the mean length $\langle A \rangle$ of attractors of critical networks. N is the number of nodes in the network.

authors	year	type	att. number	att. length
Stuart Kauffman [67]	1969	numerical	$\langle v \rangle \sim \sqrt{N}$	$\langle A \rangle \sim \sqrt{N}$
Ugo Bastolla, Gorgio Parisi [22]	1998	numerical*	$\langle v \rangle > N^x \ \forall x$	$\langle A \rangle > N^x \ \forall x$
Sven Bilke, Fredrik Sjunnesson [29]	2002	numerical	$\langle v \rangle \sim N$	
Joshua Socolar, Stuart Kauffman [116]	2003	numerical	$\langle v \rangle \gtrsim N$	
Björn Samuelsson, Carl Troein [109]	2003	analytical	$\langle v \rangle > N^x \ \forall x$	
Viktor Kaufman, Barbara Drossel et al. [72]	2005	analytical	$\langle v \rangle > N^x \ \forall x$	$\langle A \rangle > N^x \ \forall x$

* includes also analytical arguments

identical after the transient time. The number and length of the attractors of the network are not influenced by those frozen nodes. In Sec. 3 will be explained, that the attractors of the entire network can be found from combinatorial arguments incorporating some of the non-frozen nodes. As there are only few of those in the frozen phase, the attractors will therefore be short.

In the *chaotic phase*, it is $\lambda > 1$. If in the stationary state the value of one node is changed, this perturbation propagates during one time step on average to more than one other node. Even after long times there will still be nodes changing their state because of the perturbation and so there are a lot of non-frozen nodes.

The focus of Sec. 3, 5 and 6 is on the border between the two phases, on *critical* networks. It was believed that that critical networks would have neither trivial dynamics nor extremly long attractors.

The mean number and the length of the attractors in critical networks have been the subject of a number of publications, see Tab. 2.2. First results on attractor statistics predicted a square-root behaviour in the number of nodes which fit the idea that attractors correspond to cell-types and nodes correspond to genes. Increasing computer power lead to changing predictions for both the mean number and the mean length of attractors [21, 30, 116]. This numerical work was accompanied by several analytical papers. Only 30 years later did BJÖRN SAMUELSSON and CARL TROEIN [110] supply a beautiful proof that the mean number of attractors grows faster than any power law in the number of nodes N for large system sizes N.

3 Classical critical Boolean networks

In the original variant of random Boolean networks (the N-K-model), the out-links of the underlying directed graph of are to be completely random. This leads to a Poissonian degree distribution. The update is synchronous, all nodes are updated at the same time.

In the present chapter it is shown that attractor periods in large critical Boolean networks are power-law distributed. Using analytic arguments, this previously hypothesised result based on numerical evidence is explained here. The arguments are based on the method of *relevant components*.

3.1 Classical Boolean networks

The focus is on an ensemble of networks with N nodes in which the behaviour each node depends on exactly K other nodes. The Boolean states of the nodes are updated in an *synchronous* manner at uniformly spaced time steps that can be assumed to be of unit duration, $\sigma_i(t+1) = f_i\left(\sigma_{i_1}(t), \ldots, \sigma_{i_K}(t)\right)$, see Eqn. (2.4).

A particular directed graph together with the set of Boolean functions, $\mathscr{F} = \{f_1, f_2, \ldots, f_N\}$, defines a *network realization*. An ensemble of network realizations is considered. As described in Sec. 2.3.1 there can be one or more attractors of the dynamics, each of which has a basin of attraction corresponding to the region of state space from which the dynamics eventually collapses to that attractor. The late time dynamics of a Boolean network can be quantified by the number of attractors ν, their periods l_j, and the size of their basins of attraction A_j, with $j = 1, 2, \ldots, \nu$. Depending on the set of Boolean functions, $\mathscr{F}$, describing the dynamics of the nodes, the dynamics of a network ensemble can be classified into two phases with distinct behaviour. The focus lays on the boundary in between which is referred to as *critical*, compare Sec. 2.3.1. Critical networks can be obtained either by construction, or by an evolutionary process for the Boolean functions [86, 19].

3.2 Different types of nodes

It has proven useful to classify the nodes of a RBN according to their behaviour on attractors [92, 30, 116, 110, 47, 72, 23].

- The *frozen nodes* have states which become constant after some time. Interestingly, the nodes that become frozen are the same nodes for most initial conditions, and they constitute the *frozen core*. The frozen core is identified by starting from nodes with constant functions and by iteratively identifying nodes that become frozen due to frozen inputs. Networks without frozen functions can also develop a frozen core [102], however, the mechanism is different. In critical networks the frozen core comprises all but a proportion $\sim N^{-1/3}$ of nodes. A *frozen node* (like node e in Fig. 2.4) is not important for the long-term dynamics.

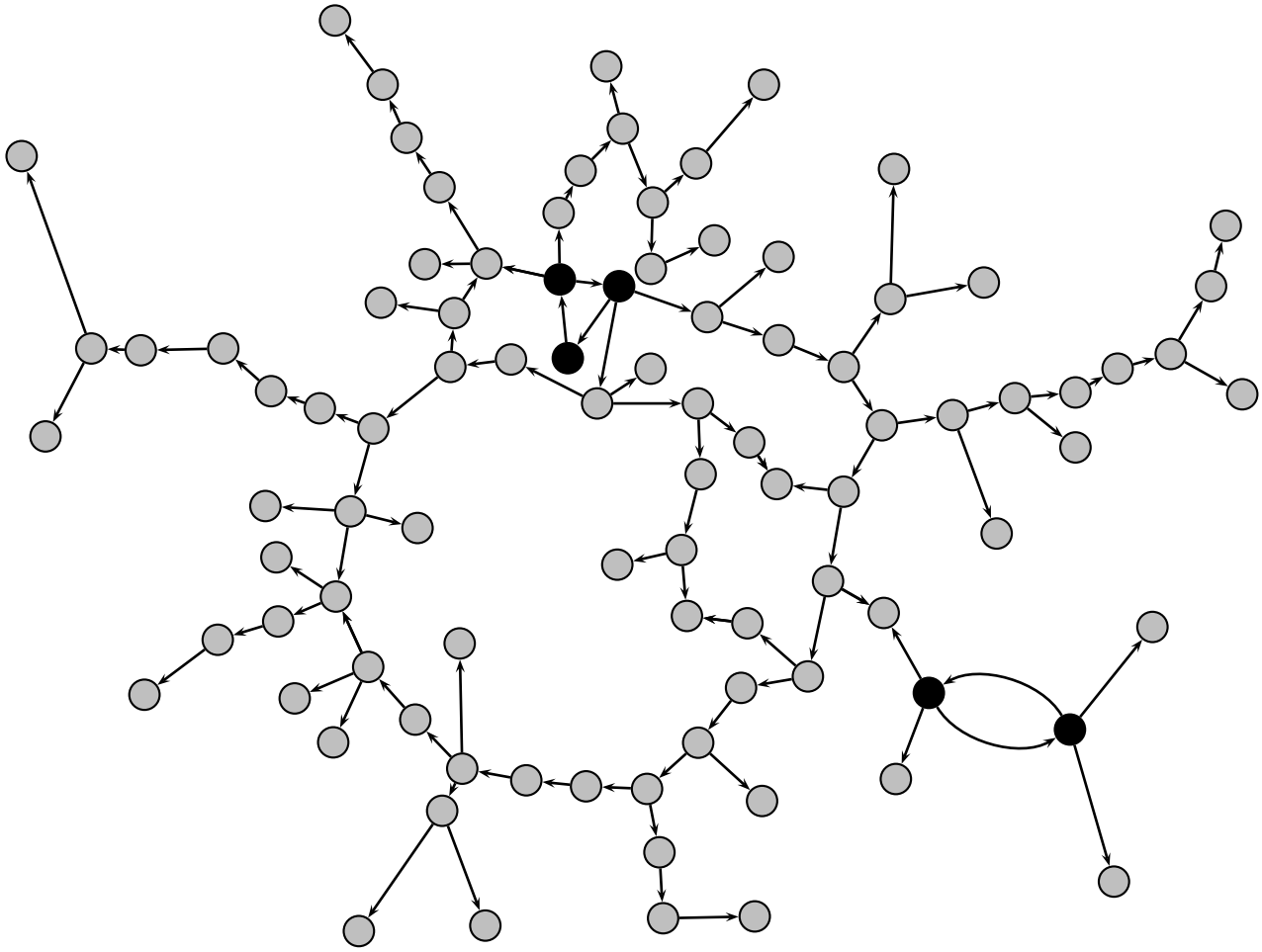


Figure 3.1: Example of a critical $k = 1$ network topology with two relevant components marked with black nodes. The links to frozen nodes have already been removed, the gray nodes are either irrelevant blinking ones or frozen, depending on the corresponding functions.

- The blinking nodes are *non-frozen nodes* and can be subdivided further:
  - The dynamics of *irrelevant nodes* (node a and b in Fig. 2.4) is completely determined by other blinking nodes. Those nodes form outgoing trees which can be cut off when searching for the nodes being essential for the attractors.
  - The *relevant nodes* (nodes c and d in Fig. 2.4) have different dynamics on different attractors. Their behaviour on the attractors is independent of those of the irrelevant ones. By definition, each relevant node influences at least s single relevant node [24]. The relevant nodes determine the attractors, and the number of relevant nodes scales in critical networks as $N^{1/3}$ [72].

The set of relevant nodes can be pooled together into components which are connected subgraphs. Each of these subgraphs is a *relevant component*, see Fig. 3.1 for an example. In critical networks, the number of relevant components scales $\sim \log(N)$ with the number of nodes N.

3.3 Concept of relevant components

The attractor period is the number of synchronous update steps needed until the same network state occurs again. All possible attractor periods of a network realization can be deduced by combinatorics from the periods of the components. Each attractor period is a *least common multiple* (LCM) of possible periods for each relevant component of the realization. For example, if there are two relevant components with possible periods $p_1 \in \{2,3,6\}$ and $p_2 \in \{1,2\}$, then the possible attractor periods are $A \in \{2,3,6,12\}$.

For critical networks, it is known that almost all relevant components are simple loops with only the largest component potentially being more complex [57]. This will now be used to consider what happens for a particular network realization starting from a particular initial condition. In a loop of l nodes every node i has exactly one input from node $(i-1)$ for $1 < i < N+1$, with the closing condition $\sigma_{N+1} \equiv \sigma_1$. In this case, the behaviour of node i is determined by the input it receives from node $(i-1)$. As the loop is a relevant component, there are no nodes having a Boolean function that gives a constant output regardless of the state of the previous node — constant outputs would block the loop and immediately lead to a fixed point attractor. Thus, two possible coupling functions are left for each node, either "copy" or "invert". According to the dynamics, one can divide loops into *even loops* and *odd loops* depending on the number of invert-functions they contain. The analysis can be reduced to the question whether or not there is an inverting Boolean function, $f_i = 1 - \sigma_{i-1}$ by the following straight-forward mapping: Choose two arbitrary nodes with an inverting function each and change that to a copy function, then flip all values of the intermediate nodes. The number and length of the attractors is not changed by this substitution.

Every synchronous update of an even loop can be imagined as an incremental rotation of the whole configuration. In an even loop after at most N updates the same configuration is reached again. In an odd loop a period of at most $2N$ is needed to obtain the same configuration.

General loops

Loops with a non-prime-number of nodes can have a shorter attractor period. For example, on an even loop (with only copy-functions) consisting of 4 nodes the pattern '0101' has a period length of 2. However, such cases of shorter periods than for prime loops can be ignored as they become less probable the more larger relevant components appear. To justify this, consider a loop of length L. Let $\mathscr{D}$ be the set of divisors of L. Shorter attractors have periods that are elements of this set. Since, as mentioned before, the two fixed point attractors with all nodes having the same values are excluded, there are $2^L - 2$ possible states for a loop of this length. How many of these possible states are realized as part of a shorter period attractor?

Clearly, if L is a prime number, then the cardinality of the set of divisors is $|\mathscr{D}| = 2$. This observable can be quickly determinded[1] and grows extremely slowly with growing L, e.g., $\max_{L\leq 10^4}(|\mathscr{D}(L)|) = 64$ and only $\max_{L\leq 10^8}(|\mathscr{D}(L)|) = 768$. Thus, the number of divisors of L is much smaller than L itself. This implies that shorter periods of a loop do not occur very often. The probability to have a period length smaller than L, $P_{<L}$, can be estimated as follows. In principle, the number of states on non-fixed point periods of the loop that are shorter than L must be counted. This number is the duration, or length, of each shorter period times the number of occurrences of that period. In the worst case, the period is $L/2$, which is only possible for even

[1] By `Max[Table[Length[Divisor[i]],{i,1000}]]` Mathematica™ evaluates $\max_{L\leq 10^4}(|\mathscr{D}(L)|)$.

number of nodes in the loop. For $|\mathscr{D}| = 3$ (only one additional divisor beside 1 and L), $P_{<L}$ is maximal, because all states of the period less than L are united at the only non-trivial divisor value, $L/2$. Thus, since there are $2^{L/2}$ different patterns of that length, an upper bound for the fraction of states in an attractor with period shorter than L is

$$P_{<L} \leq \frac{2^{L/2}}{2^L - 2} \approx 2^{-L/2}. \tag{3.1}$$

Note that the probability $P_{<L}$ vanishes for growing L. A similar argument incorporating the prime number density has been used in [48] to evaluate a lower bound for the mean attractor length in $K = 1$ Kauffman networks.

3.4 The attractor length distribution in literature

The topic of determining the average period $\langle \nu \rangle$ and average number $\langle A \rangle$ of attractors in critical RBNs has a long history. Back in 1969, Kauffman studied the mean number of attractors in critical networks with $K = 2$ inputs per nodes, he found that $\langle \nu \rangle \sim \sqrt{N}$ where N is the size of the network [67]. By then, literature suggested that the dependence of the number of cell types on the number of genes is comparable to this scaling. The number of genes was estimated by assuming that the mass of DNA is proportional to the number of genes. Nowadays, this assumption is known to be incorrect. Furthermore, the results for $\langle \nu \rangle$ and $\langle A \rangle$ in RBNs was proven wrong. Only recently it has been shown that both the average number and length of attractors in critical networks increase with network size N faster than any power law, compare Sec. 2.3.1 and Ref. [110, 46, 93]. Determining the attractor distribution, however, has received less attention.

In [28] an algorithm to study the attractor distribution of Boolean networks of up to $N \sim 10^5$ was proposed which was much beyond preceding works. They numerically found a scale-free behaviour for networks with $K = 2$ and bias $\rho = 0.5$ (that is the probability to have a "1" as the value of a Boolean function). Other researchers used a probabilistic approach in the framework of the annealed approximation [23, 21, 20]. The modular structure of Kauffman networks was mostly numerically there (in particular for $K = 4, \rho = 0.25$) and the attractor lengths distribution was found to be very broad on the critical line, having approximately the shape of a power law. Then, a phenomenological method method was proposed [5], stating that the attractor length as a function of the system size can be split up (Eq. (7) in [5]) into the clustering of the nodes into islands of active nodes, a function for the cluster size, and its period length and the relationship between those periods to the total period of the system. Those three constituents can then be approximated numerically. Another study worth mentioning looks at the transient time of the attractors which strongly influences the time scale needed to determine the attractor size distribution of a given network [79].

Recently, it was found that the *evolution* of Boolean networks leads to a a stationary self-organised state, the attractor distribution was measured as an independent outcome of the evolution: Boolean agents play a minority game[2] on the network [100]. The distribution of attractor lengths in the stationary state shows a power-law behaviour for large enough attractor lengths. Similarly, the coevolution of network structure and network dynamics evolves to

[2] The minority game is a model where agents compete through adaptation for a finite resource, players being in the minority side win a turn of the game.

networks exhibiting a power law attractor distribution[86]: The topology-evolving rule there is that a frozen node grows a link while an active node (changing its value) loses a link.

It is (computationally) easy to generate the attractor distributions only for small sized networks and many authors have done so, see references within [47]. In the present chapter an analytical understanding for all kinds of critical networks is offered, explaining how this behaviour arises.

3.5 Calculating the attractor distribution

In order to calculate the attractor distribution of a critical network, we start by looking at the topology of such a network, more precisely at the distribution of relevant component loops. An essential result is that the distribution is Poissonian. Thus, it is known that such a loop consisting of l nodes appears with probability $p_l = l^{-1}$ for loop-lengths smaller than a cutoff, $l < l_{\max}$. The cut-off length depends on the size of the network, $l_{\max} \sim N^{1/3}$ [46].

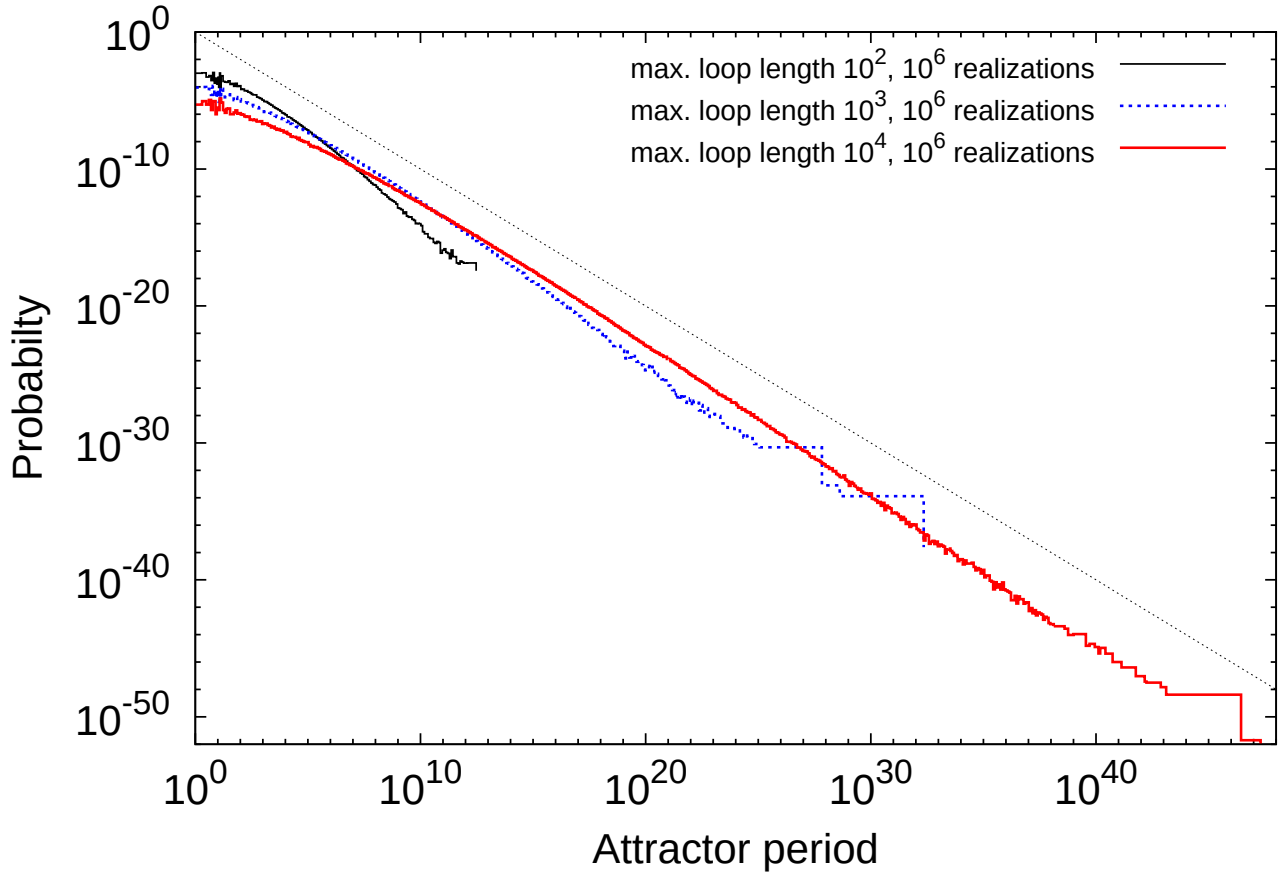


Figure 3.2: Distribution of attractor periods l. These probability distributions are found by generating relevant components and deducing the corresponding attractor period for 10^6 realizations. The known loop-distribution is used, up to a cutoff value. The figure shows different cut-offs of 10^2 (blue dashed line), 10^3 (red solid line) and 10^4 (solid black line). It is assumed that each loop contributes only to one attractor that in turn has an attractor period comparable to its number of nodes. The dotted line corresponds to a power-law with exponent -1.

Equipped with this knowledge one can move on to the dynamics. The period of the attractor is determined by the LCM of all size relevant components. As discussed above, it is known that, in the large network limit, only the largest relevant component of a critical network has a finite probability of being more complex than a simple connection-loop [71].

One now makes the approximation that *all* relevant components are simple loops. This assumption is allowed when neglecting the very long periods which appear with a small probability. The complex components can be constructed from additional links in loops or by interconnecting multiple loops. Depending on the network topology and on the Boolean functions at the nodes with more than one relevant input, the period of the complex component can be either smaller or larger than the length of the loops which are used to construct it, for details see [71]. Shorter period components will be included by the random procedure by just picking smaller loop lengths. The validity of this assumption improves as $l_{\max}$ increases.

It is assumed that the period of the attractor corresponding to a given loop is equal to the length of the loop. This assumption is even made for loops with odd length that actually have a period that is twice as long. However, the factor 2 that would appear for odd length loops, but for estimating the order of magnitude this does not significantly change the conclusions.

Next, using numerical sampling the validity of the arguments is explored. Each of the s different samples corresponds to a single attractor of a network realization. Note, that each network realization might have more than one attractor, but only one per realization is taken. The procedure described above is implemented by taking a set of loops of length l, each of which occur with probability $1/l$, and by calculating the LCM. Using this method one ends up with a histogram of attractor periods, see Fig. 3.2.

In order to display the histogram data in a logarithmic representation, the data is binned, i.e., attractor lengths within a certain range are put into one bin. The width of the bins grows with a binning factor, a given bin has b times the length of its neighbouring bin on the left. The results are shown with a binning factor of $b = 1.2$ for $l > 14$, for small l just the length of the attractor itself is used. By this choice it is guaranteed that each bin contains at least one attractor length. Binning is used when the bins may contain more than one attractor length, i.e., for attractors with more 14 states. The histogram is normalised such that the total probability is one. As expected, as the network size, or the maximum loop length, grows the distribution approaches a power law with exponent -1. Note that with this new sampling method huge system sizes can be studied. The free parameter in the method is the cutoff $l_{\max}$ which is a function of the system size. A further simplification is to take just the product of the individual loop periods instead of the LCM.

Divisibility effects

There are two ways for considering what happens when the attractor period is short. First, the LCM method described above will be applied and, second, the product of the loop lengths of all loops instead of the LCM is taken. Qualitatively, the approximated functional form of the attractor period distribution using both methods is similar. However, there are some differences. Most notably, using the LCM method, the distributions converge more slowly to the $1/\nu$-scaling expected for large networks. This is because the LCM gives a more accurate estimate.

The spikes in the distribution of attractor periods, see Fig. 3.2, are due to the number of divisors, which can be seen from Fig. 3.3. The histogram for the distribution of attractor periods as obtained by an evolution of a Boolean model, shown in Fig. 2 of [100], exhibits a very similar

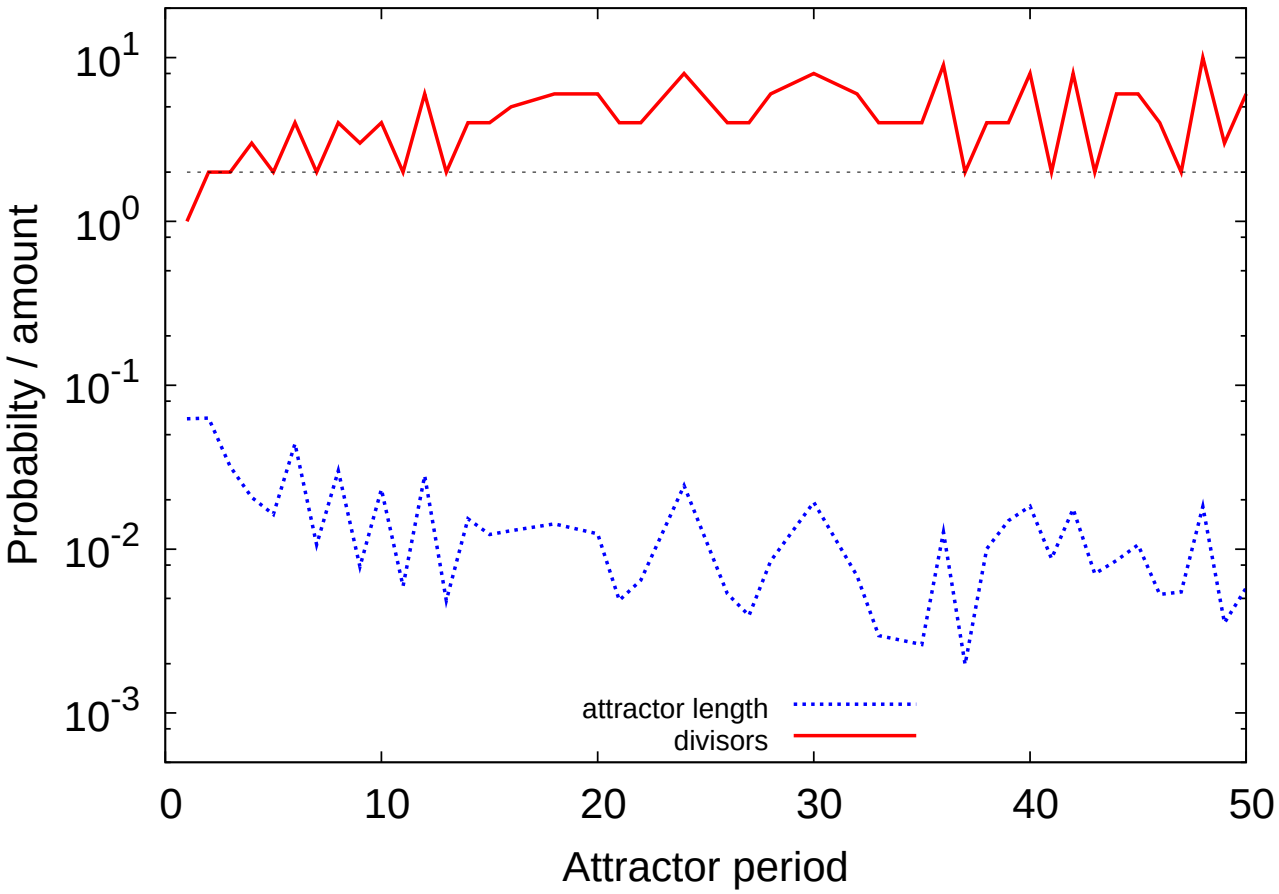


Figure 3.3: Probability of attractors to have period l (blue dashed curve) and the number of divisors of l (red solid curve). Each time the values in the upper curve have the value 2 (dotted horizontal line), l is a prime number.

spike-pattern as observed in the present work. Note that in that paper binning was used for smaller attractor periods, thus some peaks are averaged away.

As explained above, a given attractor length can be approximated by taking the least common multiple of different loop lengths (as proxy for the loop period) and beside that the probability for each loop length is known. If *all* loops up to certain length x contribute to the attractor, number theory provides a formula for that. Using the prime number theorem, and an inequality proved by Nair [121] the following result holds for the least common multiples of the first x positive integers with $x = \{1, 2, \ldots\}$

$$s(x) \equiv \mathtt{LCM}(1, 2, \ldots, x) \geq \exp\left(x\,(1 + \mathcal{O}(1))\right) \tag{3.2}$$

The series $s(x)$ starts with $1, 2, 6, 12, 60, 420, \ldots$ [115] and represents all possible attractor periods. For large x, the probability for an attractor period which is constructed of *all* possible periods lengths is negligible. The reason is the e^x in Eq. (3.2) which appears in the denominator of the probabilities for the overall attractor period. For smaller x of say $x < 10^3$, the approximation (3.2) from number theory does not hold yet, but then the approximation $p(l) \propto 1/l$ does hold.

Problems of direct simulation

Only a full state space enumeration of the dynamics allows one to obtain the exact attractor distribution. However, this is only possible for small system sizes. This is true even if an intelligent pruning algorithm is used that disregards irrelevant nodes and simulates only the dynamics of the relevant nodes for a given realization.

For this reason, previous studies of the attractor period distribution have relied on sampling. However, sampling has potential problems. One problem that can occur when generating attractor-statistics by sampling of various network realizations is undersampling. Undersampling occurs when simulating without any prior knowledge about the structure of the state space. If nothing about the relevant components is known, then in principle one has to determine to which attractor each initial condition converges to. In order to do this, one has to determine the successor for each state, meaning 2^N updates. Because of this restriction, it is only possible to sample only a set of initial configurations that correspond to a negligible fraction of the state space for large system sizes. Another known problem that occurs with sampling is that the frequency with which attractors are found depends on the size of their basin of attraction, as mentioned in e.g. [110].

The new method has neither the problem of undersampling nor of being biased by the basin sizes, and allows to effectively study very large networks. It constructs the overall attractor of a network realization by taking the period of the relevant components the realization is constructed of. The analytic arguments explain the numerical evidence found by others that attractor periods in large critical Boolean networks are power-law distributed. Thus, critical Boolean networks exhibit scaling also in the attractor distribution, a property that until now has not been analytically shown.

3.6 Summary

In this chapter results for the attractor distribution for critical Boolean networks are presented, the work is carried out in cooperation with with KEVIN BASSLER and will be submitted soon.

The method of relevant components is explained and it is shown how the attractor distributions for critical random Boolean networks matches the one occurring in Boolean networks evolved to criticality.

4 Varying the updating scheme

The usual synchronous way of updating is not very realistic [61] as natural systems are rarely controlled by an external clock. It is known that properties of attractors for synchronous dynamics can differ from those for asynchronous dynamics. For instance, for cellular automata part of the self-organisation is an artifact of the synchronous updating [66]. It is possible to distinguish at least four kinds of different updating schemes for random Boolean networks which change the dynamics considerably:

- Classical Random Boolean Networks (CRBNs) have been considered in detail in the previous chapters, the nodes are updated synchronously.
- The updating times may only slightly deviate from the synchronous pace due to *fluctuations in the updating times* of the nodes, see Sec. 4.1.
- The *deterministic asynchronous* update refers to some kind of "quenched-randomness" in the updating times. The updating order of the nodes is defined once per realization and then stays fixed, see Sec. 4.3. The corresponding ensemble shall be abbreviated as Deterministic Random Boolean Networks (DRBNs).
- The *asynchronous stochastic* scheme is the extreme case where a single node to be updated is picked at random, see Sec. 4.2. Such a model is often called Asynchronous RBN (ARBN) [54, 53]. ARBNs are mostly studied numerically with the focus on various measures of stability as defined in [101, 108, 60, 91, 88].

The fixed points do not depend on the updating scheme as a mapping a state vector onto itself requires that the values of all nodes stay the same, no matter in which order they are updated. The minimal number of attractors is naturally determined by the number of fixed points. Their maximum number can be approximated using the method of relevant components, this has been performed in a mathematically rigorous way in [17].

In the present chapter it will be shown mostly analytically that the number of attractors changes completely when going from CRBNs to ARBNs and DRBNs. To a large extend, the present section follows the discussion which has been published in cooperation with Barbara Drossel [57, 59].

4.1 Fluctuations in the updating time

Konstantin Klemm and Stefan Bornholdt analyse a model with fluctuating updating times of the nodes [77, 76]. They replace the discrete update times as known from the synchronous case by a continuous time variable. The dynamics is only slightly desynchronised by shifting the individual updates of nodes to earlier or later time points, $t \pm \varepsilon$ with a small $\varepsilon > 0$.

To visualise the effect of such an updating scheme, let the system be a feedback loop consisting of three nodes, see Fig. 4.1. All Boolean functions are chosen to be negations, i.e., each node inverts the value of its input, $\sigma_{i+1}(t+1) = 1 - \sigma_i(t)$ for $i \in \{1,2,3\}$ with $\sigma_4 := \sigma_1$. The

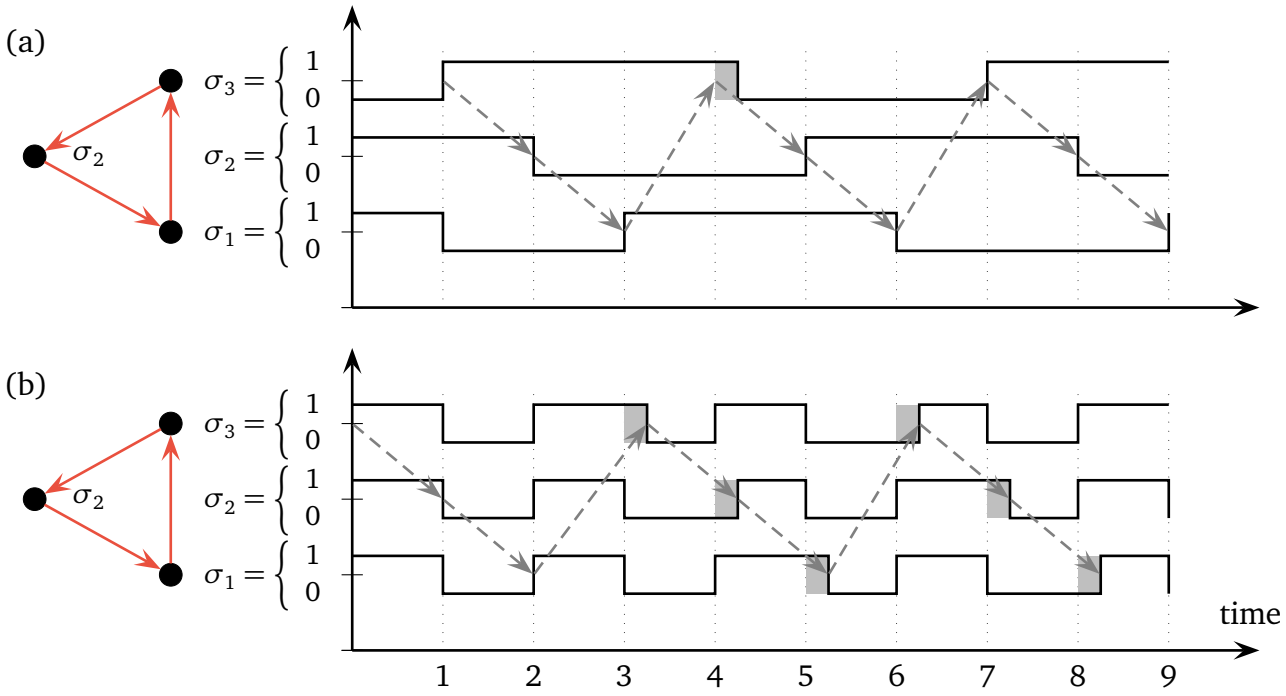


Figure 4.1: A stable attractor (a) and an unstable attractor (b) under timing fluctuation of a feedback loop. The dotted vertical lines are the paces of a synchronous metronome, the fluctuations are depicted by gray shaded boxes. The criterion for the attractor type is whether or not the fluctuations (shaded boxes) die out.

time evolution of a *stable attractor* (against timing fluctuations) shows a unique causal chain of flipping events, marked by dashed arrows in Fig. 4.1. A retarded update does not affect subsequent updates. In contrast to that, on an *unstable attractor* the system does not regain synchronicity after a perturbation. The system Fig. 4.1 consists of three independent chains of flipping events, one of these chains is indicated by dashed arrows. Retarding an update affects subsequent events in the same causal chain only.

4.2 Asynchronous stochastic updating

For stochastic asynchronicity, at each time step a single node i is chosen at random to be updated according to its Boolean function f_i. It will be argued why the mean number of attractors in a critical Boolean network under asynchronous stochastic update grows like a power law and that the mean size of the attractors increases as a stretched exponential with the system size N,

$$\langle A \rangle \sim \exp(N^{\alpha}) \qquad \text{for some fixed } \alpha. \tag{4.1}$$

This is in strong contrast to the synchronous case, where the number of attractors grows faster than any power law. The following considerations are part of Florian Greil's diploma thesis and are published in [59].

As the dynamics is no longer deterministic, an appropriate definition of an attractor must be given. As used e.g. in [61], a *loose attractor* is a subset of the configuration space such that for every pair of configurations on the attractor there exists a sequence of updates that leads from one configuration to the other.

Loops under asynchronous stochastic update

The non-frozen nodes of critical networks essentially form a $K = 1$ network, i.e., all but one input of a non-frozen node are frozen. In critical networks, relevant nodes are arranged in $\mathcal{O}(lnN)$ components, most of which are simple loops. Typically, there is only one component that is not a simple loop [70]. The simplest relevant component is a loop consisting of nodes with $K = 1$ incoming edges and Boolean functions which either "copy" or "invert" the previous node's value.

For synchronous update, each configuration in a connection loop is on a cycle in configuration space and occurs again at most after N $(2N)$ time steps for even (odd) loops, see Sec. 3. The number of cycles increases therefore exponentially with N. In contrast, most configurations are transient the asynchronous case and as such not part of any attractor. At the end, only two attractors per even asynchronous loop and one for and odd asynchronous loop are left. The reason for this is that a *domain* of neighbouring nodes that have the same value increases or decreases with probability $1/N$ per computational step. The domain size therefore performs a random walk, and for an even loop no domain is left after of the order of N^3 updates. The attractors are the two fixed points of the system with all nodes switched on or all being off. For an odd loop, the nodes of a domain change their state at the node with the inverting Boolean function, and the total number of domain walls is therefore odd. The attractor contains only one domain wall that moves around the loop, and the attractor comprises $2N$ configurations. The dynamics on such a loop is closely related to the Glauber dynamics [55] of a one-dimensional Ising chain with cyclic boundary condition at temperature $T = 0$, where the domains also shrink and grow with a fixed rate and where the equal-time correlation function obeys a scaling form $C(r,t) = f(r^2t^{-1})$ [33]. The dynamics of an odd loop can be mapped onto the dynamics of an Ising chain with one negative coupling. It is a frustrated system in which not all bonds can be satisfied simultaneously. To conclude, by going from synchronous to asynchronous update, the number of attractors of a loop is reduced from an exponentially large number to 1 or 2. This was also pointed out in [78] for asynchroncy due to fluctuations in the updating times.

Critical $k = 1$ networks

Next, critical networks with one input per node, $k = 1$, are considered. The Boolean coupling functions are again "copy", $f_i = \sigma_{i-1}$, and "invert", $f_i = 1 - \sigma_{i-1}$. Under asynchronous stochastic update, each loop has at most two attractors. Critical $k = 1$ networks consist of loops and outgoing trees attached to them, see Fig. 3.1. The nodes on trees rooted in even loops are frozen because the loop is on a fixed point. The nodes on trees rooted in odd loops can assume any combination of states, since one can find to each possible state of a tree a sequence of updates that generates it. The mean number of attractors, $\langle \nu \rangle$, of networks with n relevant loops can be calculated by summing up the probability of an odd loop to be on an attractor which is not a fixed point. As each loop is even or odd with equal probability, it is possible to approximate $n \sim \ln N/2$. Thus,

$$\langle \nu \rangle = \frac{1}{2^n} \sum_i \binom{n}{i} 2^i \quad = \quad \left(\frac{3}{2}\right)^n \simeq \left(\frac{3}{2}\right)^{\ln(N/2)}, \tag{4.2}$$

which is a power law in N. The number of nodes in trees is of the order of N. On average, half of the trees are rooted in odd loops. Consequently the mean attractor, $\langle A \rangle$ size increases exponentially with N for $k = 1$ networks.

Critical $k = 2$ networks

Finally, the most frequently studied critical networks with connectivity $k = 2$ are analysed. Each of the $2^{2^k} = 16$ possible Boolean coupling functions is chosen with equal probability. Apart from the order of $N^{2/3}$ nodes all nodes are frozen, the number of relevant nodes scales as $N^{1/3}$ [116], and only a fraction $N^{-1/3}$ of these relevant nodes have two relevant inputs [72, 47]. Consequently, the proportion $N^{-1/3}$ of non-frozen nodes (whether they are relevant or not) have two non-frozen inputs. The other relevant nodes have one relevant input (as the second input comes from a frozen node). The remaining non-frozen nodes (of the order of $N^{2/3}$) are on trees rooted in relevant nodes. Just as for the $k = 1$ networks, there are of the order of $\ln(N)$ independent relevant components [23]. In contrast to the $k = 1$ networks, these components are not always simple loops, but may contain several nodes with two relevant inputs. In order to obtain results for the number of attractors of the networks, the attractors of such relevant components have to be investigated now.

Relevant components beyond loops

Let μ nodes of a component have two inputs. Beside loops ($\mu = 0$) there are two components with $\mu = 1$ and then more complicated components with $\mu > 1$. If $\mu = 1$ there are either two loops with a cross-link (○–○-component) or a loop with an extra link (⊘-component), see Fig. 4.2. It is known that the ○–○-component appears twice as often as the ⊘-component [70]. The dynamics under synchronous update of such a system is also studied there, the number of attractors in both systems increases exponentially with the number of nodes. With asynchronous stochastic update the number of attractors becomes very small.

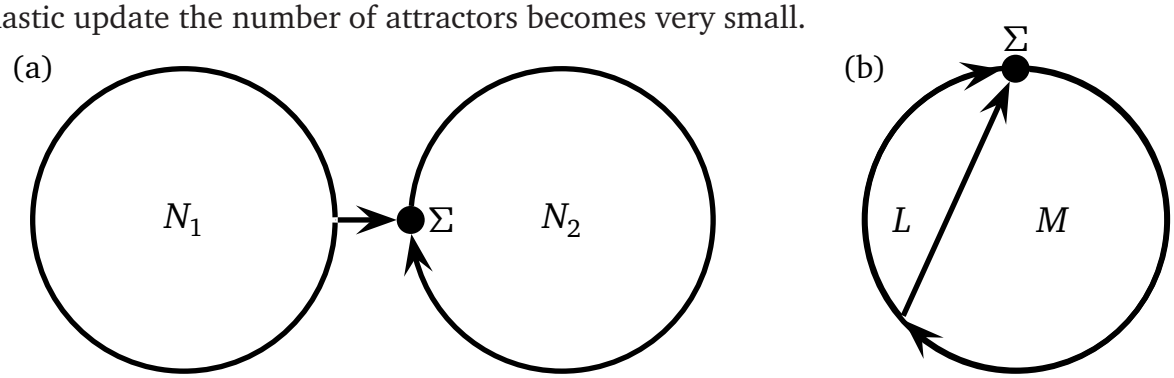


Figure 4.2: Two simple components beyond loops. The ○–○-component (a) consists of a loop with N_1 nodes connected to a loop with N_2 nodes. The ⊘-component (b) is a loop of $N = L + M + 2$ nodes having an additional link. In both cases node Σ depends on two inputs, $\sigma(\Sigma) = f_\Sigma\left((\sigma(G_1), \sigma(G_2)\right)$.

For the case of the ○–○-component, Fig. 4.2(a), the left loop is independent of the right loop. The attractor of the ○–○-component is either a fixed point (if the loop is even), or it has one domain wall moving around the loop. That implies two cases: If the first loop is on a fixed point, it provides a constant input to the second loop, which therefore behaves like an even loop, an odd loop, or a frozen loop. In other words, the system can have at most three attractors. If the first loop is odd, if provides a changing input to the second loop, which can therefore have an attractor that contains an arbitrary and fluctuating number of domain walls. Consequently, a loop that has one external input can show one out of four different types of behaviour on an attractor:

1. The loop can be at a fixed point with $\sigma_i \equiv 0$ for all nodes i.

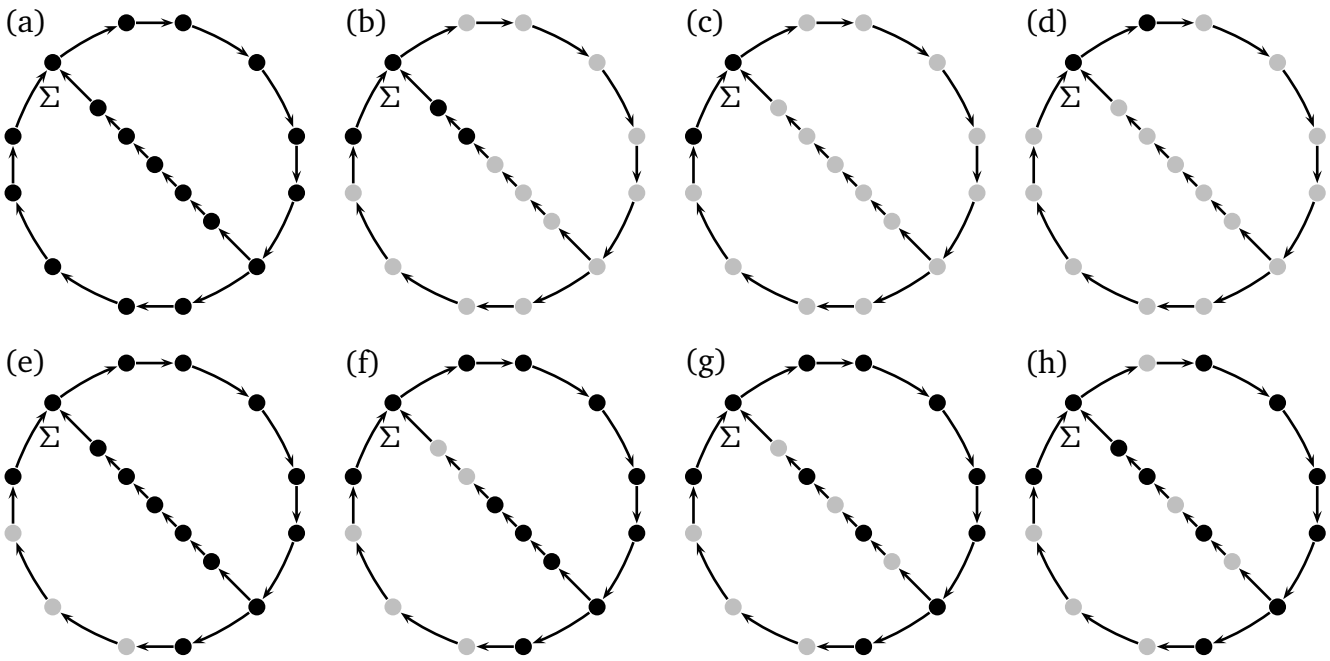


Figure 4.3: Procedure to create new domain walls on the ⊘-component. The Boolean functions of all nodes with one input are chosen to be "copy". The coupling function of the node Σ with two inputs is such that only two "off" inputs (black nodes) lead to a "on" (light gray) output, all other combination give dark output. Starting from (a) the only dynamics that can occur is the generation of a 1 at the node Σ and afterwards a whole domain of nodes of 1s (b). In case the random update provides configuration (c) next, a new 0 can be created as output for Σ as sketched in (d). This domain might grow until state (e) is reached. An update sequence similar to (a–e) allows to generate another domain of 1s which is now (f) located in the chain of nodes which distinguish the branch of the component of the ⊘-component from a loop. Abstracting further, any amount of domains of 1s can be created. In (g), there are 4 domains of 1s in the system, from which exists a trajectory to (h), a configuration where one domain passed through node Σ.

2. The loop can be at a fixed point $\sigma_i \equiv 1 \ \forall i$.
3. There is exactly one domain wall which moves around the loop.
4. The number of domain walls in the loop fluctuates.

Without loss of generality, it has been assumed that all coupling functions for nodes with one input are "copy", $f_i = \sigma_{i-1}$. These four types of behaviour will become important later.

For the ⊘-component, Fig. 4.2(b), it is again assumed that all coupling functions for nodes with one input are "copy". If this component has one fixed point, it is the only attractor. If the component has two fixed points, those are also the only two attractors. This is because one can reach a fixed point from an arbitrary initial state by updating one node after another by going around the loop in the direction of the links. After at most two rounds the fixed point is reached. Only if the coupling function for the node with two inputs has no fixed point, a more complicated attractor occurs. Without loss of generality, this function is defined to have the output 1 if and only if both inputs are 0. By considering the possible update sequences, one finds that the component can accumulate a large and fluctuating number of domain walls, the procedure leading to new domain walls is illustrated in Fig. 4.3.

Relevant components with several nodes with two inputs

Equipped with the results for relevant components with $\mu = 1$ nodes with two inputs, components with $\mu > 1$ nodes with two inputs are the next challenge. A *section of nodes s* shall be defined to be a sequence of nodes starting at a node with two inputs and ending right before another such node. A section can branch and have several end points. Clearly, the number of sections is the number μ of nodes with 2 inputs if there are any, a simple loop is counted as one section, thus

$$s := \max(1, \mu) \quad \mu \text{ counts nodes with 2 inputs in the component} \tag{4.3}$$

A section is controlled by its first node, which is the one with two inputs. Just as for the loop with one external input, a section can show on an attractor one out of the four different types of behaviour listed above. The reason is that all network configurations having more than a single domain wall in a given section must be part of the same attractor. This can be shown by the following argument: Assume that on an attractor there occur two domain walls in a section. The two domain walls can be destroyed by updating all nodes between the two walls, such that the domain enclosed by the walls vanishes. A configuration with no wall on the section (and with the state of all other sections unmodified) is therefore also part of the attractor, and consequently there exists a way back to the configuration with two domain walls on this section. By repeating the same sequence of updates, every even number of domain walls can be created in this section, and odd numbers can be created by moving one domain wall out of the section. An upper bound for the number of attractors of the component is therefore given by

$$\nu_{\mathrm{ARBN}} \leq \nu_{\max} \leq 4^s \tag{4.4}$$

with $\nu_{\max}$ is the maximal occurring attractor length when sampling a given ensemble of networks, compare Fig. 4.4.

Numerical confirmation of the upper bound for the number of attractors

This analytical result can be checked by simulations. In order to make sure that all attractors are taken into consideration, a complete search of the state space was performed, this can of course only be done for small networks. Starting from an initial state, N^3 updates are performed before assuming that the system is on an attractor. The results do not change when the length of the initial time period is varied. All states that can be reached from this last state are on the same attractor as this state. All other states that have been visited are marked as transient states. Then, an unvisited state is taken as a new initial condition in order to identify further transient states and attractors. The relevant components are constructed by starting with one loop of a certain size, and by iteratively inserting additional connections between two randomly chosen nodes.

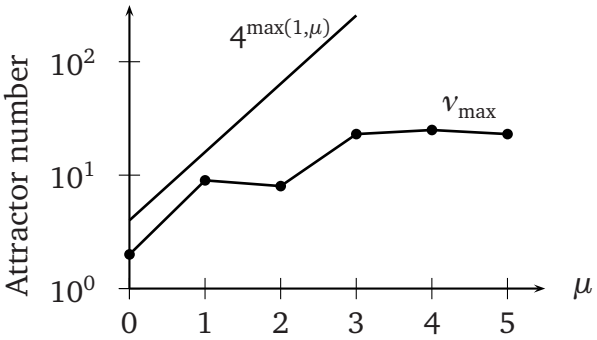


Figure 4.4: Numerical test for $\nu_{\max}$, the maximal number of attractor in ARBNs. Networks with up to $N = 17$ have been probed.

Each new connection contains a randomly chosen number of 1 to 4 nodes, such that a section may contain two domain walls in its interior. In these networks the number of sections, s, is identical to the number of nodes with 2 inputs, μ. After each insertion the number of attractors is evaluated for different choices of the Boolean coupling functions at the node with two inputs. This procedure was repeated more than 750 000 times. Networks with lower μ are probed more often. For higher μ only short links can be added without exceeding the maximal computationally possible N, there is just no node left which not already has two inputs. The largest number of attractors, $\nu_{\max}$, found for an ensemble of system with same μ is shown in Fig. 4.4.

4.2.1 Results for critical ARBNs with $k = 2$

Putting everything together, a network consisting of the order of $\ln(N)$ relevant components, with component i having μ_i nodes with 2 inputs, cannot have more than

$$\nu = 4^{\max(1,\mu_1)} \cdot 4^{\max(1,\mu_2)} \cdot \ldots \cdot 4^{\max(1,\mu_{\ln N})} \leq 4^{\ln(N)+\mu} \tag{4.5}$$

attractors. This is a power law in N if the probability distribution for the value of μ becomes independent of N for large N. Indeed, as mentioned above, each of the $N^{1/3}$ relevant nodes has two (randomly chosen) relevant inputs with probability $aN^{-1/3}$ with some constant a. Since this probability is independent for different nodes, the value of μ is distributed for large N according to a Poisson distribution with mean value a.

There are of the order of $N^{2/3}$ nodes on the trees rooted in the relevant components. These nodes can adopt any configuration if the node they are rooted in can switch its state on an attractor. Since a non-vanishing fraction of all relevant nodes switch their states on an attractor, the size of the attractor is of the order of

$$A \sim 2^{N^{2/3}} = \exp\left(N^{2/3} \ln 2\right) . \tag{4.6}$$

The size of the attractors grows like a stretched exponential function, and therefore faster than any power law.

Many of the results hold also for other kinds of stochastic asynchronous update, for instance if a certain (small) fraction of nodes is updated at each step, or if the time interval between two updates of a node is peaked at a value τ and Gaussian distributed around it.[1] In these modified stochastic models, domain walls on an isolated loop can be annihilated, but cannot be created again, leading to the same attractors as with the completely stochastic update. However, the state of the trees rooted in the loops will be dominated by a few domain walls when the distribution of update times becomes narrow, with states with more domain walls occurring rarely. Similarly, relevant loops that receive input from outside, and relevant components with nodes with two inputs will have attractors dominated by few domain walls, and the actual size of the attractors becomes in the thermodynamic limit $N \to \infty$ smaller than the size obtained by considering any possible sequence of updates.

[1] The case of Gaussian distributed update fluctuations describes an ARBN with large N, when the network of relevant nodes is coarse-grained such that of the order of $N^{1/3}$ neighbouring nodes are replaced by a single node that receives a delayed input from the previous node.

4.3 A deterministic asynchronous updating scheme

In contrast to stochastic updating schemes, *deterministic* updating schemes that are not fully synchronous have been investigated less. In [53], different asynchronous updating schemes are compared numerically among each other for small network sizes with the conclusion that the number of attractors does not depend as much on the synchroncity of the updates, as they depend on their determinism.

One way to implement a deterministic asynchronous updating scheme is be to assign a time delay to each link between two nodes. This can be motivated biologically, since the expression of genes is not an instantaneous process, but the transcription of DNA and the transport of enzymes may take from milliseconds up to a few seconds. However, a network with different delay times can be mapped on a network with synchronous update by introducing a a chain of nodes with unit delay times between nodes with longer delays. For this reason, the present work focuses on a different type of asynchronous deterministic networks, namely on those with a "quenched randomness" in the updating times. This means that some nodes are less frequently updated than others. Such networks represent an intermediate step between synchronous and fully stochastic updating rules. Quenched randomness in update times can be motivated by the necessity for each node to have a *refractory time*, i.e., a time during which a node cannot respond to an input signal because it is still processing or recovering from the previous input signal. While a realistic implementation of update delays and refractory times would lead to more complex rules, using fixed update intervals for each node is an instructive and important step in the investigation of not fully synchronous networks.

Defining node-based delays

In the classical synchronous case, the assignment of connections and functions to each node remains fixed throughout the time evolution, this causes the model to be called *quenched*. Two new parameters per node are required, they are also fixed for a given realization.

1. To each node i a *delay time* τ_i is assigned, which is fixed for a given network realization. The value σ_i of node i is updated in time intervals τ_i. This defines the internal time scale of the node, i.e., how long a given node does not respond to its inputs, see Fig. 4.5.

2. Then, each node is assigned an initial *phase* φ_i. This phase determines the time to the first update of node i, therefore $0 \leq \varphi_i < \tau_i$ holds.

The system is deterministic as the succession of network states $\boldsymbol{\sigma}(t)$ is entirely defined by the initial condition $\boldsymbol{\sigma}(0)$ and the initial phases $\{\varphi_i\}$. The classical case of parallel update as described in Chapter 3 is a special DRBN with all $\tau_i \equiv 1, \varphi_i \equiv 0$. The size of the state space Ω generalises from $|\Omega|_{\text{CRBN}} = 2^N$ to $|\Omega|_{\text{DRBN}} = \prod_i 2^{\tau_i}$.

Approach to study deterministic dynamics

Again, the number and length of attractors is of interest. Both quantities are affected by the chosen updating scheme. It is important to realize that the classification in *frozen*, *non-frozen* and *relevant* nodes is independent of the updating scheme. To which class a node belongs, depends only on the topology of the network, and on the functions assigned to the nodes. The update time of a node does not influence the question whether a node freezes if some of its inputs are frozen. For this reason, the relevant nodes determine the long-term dynamics of

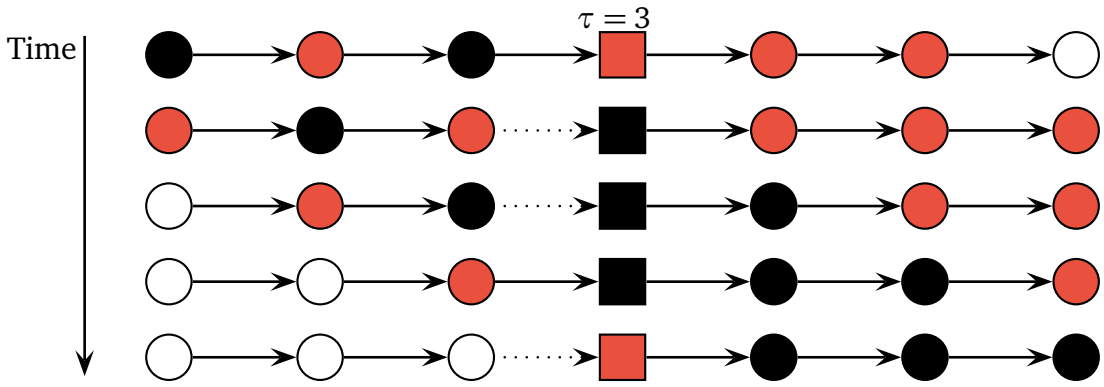


Figure 4.5: Effect of a node-based delay on a chain of nodes with Boolean function "copy". The squared node has a delay $\tau = 3$. Each node's value is either 0 (black), 1 (red) or the value does not matter (white). A dotted arrow symbolises that during a given time step the squared node does not react to its input.

random Boolean networks for any updating scheme. The dynamics of the outgoing trees is slaved by the relevant parts, and the frozen nodes obviously are irrelevant. Thus, the properties of only the relevant components under deterministic asynchronous updating will be studied in order to apply the findings about their dynamics to the dynamics of the entire network.

The further outline of the present chapter is as follows: First, simple loops with one delayed node are analysed (Sec. 4.3.2), then simple loops with several delayed nodes are taken into account in Sec. 4.3.3. Next, two loops with a cross-link and one delayed node are analysed in Sec. 4.3.4 while Sec. 4.3.5 is dedicated to a loop with one additional link and a delayed node. The chapter closes with a discussion of the consequences for the entire network in Sec. 4.3.6.

4.3.1 The concept of effective nodes

Once again, the starting point is to analyse the dynamics on a loop, where each node just copies the value of its predecessor. For all kinds of updating schemes there are two fixed points for an even loop, namely $\boldsymbol{\sigma} \in \{\mathbf{0}, \mathbf{1}\}$.

Moving towards a more general updating scheme a coarse-grained concept of *effective nodes* make senses. After some transient time only certain blocks of successive nodes will be able to change their value coherently. Let the term *block* refer to the smallest possible set of neighbouring nodes having the same values. A *domain* contains at least one block and is terminated on both sides by *domain walls* where it connects to blocks of the opposite node value. Finally, a *block candidate* is a set of neighbouring nodes with exactly as many nodes as a block but the nodes do not necessarily have the same values. The latter definition will be useful for more complex components.

4.3.2 Loops with one delayed node

The first step is to delay just a single node on a loop. Let node 1 be delayed and the lag time τ^* is assumed to be a natural number, $1 < \tau_1 \equiv \tau^* \in \mathbb{N}$, while $\tau_i = 1$ for $i > 1$. Vividly speaking, one gene needs much longer to be expressed than all the others. Since node 1 remains at the same value for τ^* time steps, node 2 receives τ^* times the same input, leading to blocks of size τ^* travelling around the loop. When the head of a block arrives at node N (which is connected to

node 1 again), node N will have the value of this block for τ^* time steps, and during one of these steps node 1 will be updated. In the following, even and odd loops will be treated separately.

Even loops with one node delayed by τ^*

Even loops (abbreviated $\oplus$) have two fixed points. The other attractors are characterised by blocks of length τ^* travelling around the loop. Assume node 1 is updated at time 0. A block with the value of node 1 will start travelling around the loop, and the head of the block will arrive at node N at time N. The next update of node 1 will be at time $T = \lfloor (N-1+\tau^*)/\tau^* \rfloor \cdot \tau^*$, where the Gaussian brackets $\lfloor x \rfloor$ denote the largest integer less or equal x. The expression in parenthesis specifies an effective time delay. The value of node 1 becomes the same as at time 0, and the same block travels around the loop again. The same consideration can be made for all starting times that are multiples of τ^*, leading to the result that the state of the loop is repeated every T time steps.

After the transient time N, the loop has reached an attractor that contains $\xi = T/\tau^*$ blocks, each of which either has only values 0 or values 1, and with the block that contains node 1 being shorter than the other ones, if N is not a multiple of τ^*. Every attractor corresponds to a pattern of blocks travelling around the loop. If the number of blocks is a prime number, $\xi \in \mathbb{P}$, the length $A_\oplus$ of the attractors is identical to T. The block has to wait at $(N-1)$ nodes for 1 time unit and then it has once to wait for τ^* time units,

$$A_\oplus = T = \left\lfloor \frac{(N-1)\cdot 1 + 1\cdot\tau^*}{\tau^*} \right\rfloor \cdot \tau^* = \xi \cdot \tau^* \qquad \text{for prime number of blocks, } \xi \in \mathbb{P}. \tag{4.7}$$

For $\xi \notin \mathbb{P}$, the pattern of blocks can have a period that is a divisor of ξ, in which case the attractor length is shorter.

The number of different attractors for an even loop, $\nu_\oplus$, can be calculated from the number of different patterns, 2^ξ. Each of those patterns corresponds to one attractor. Including the 2 fixed points this leads to

$$\nu_\oplus = \frac{2^\xi - 2}{\xi} + 2 \qquad \text{for prime number of blocks, } \xi \in \mathbb{P}. \tag{4.8}$$

If the number of blocks is not prime, $\xi \notin \mathbb{P}$, the number of attractors increases as the length of some attractors is shorter. For $\tau^* \geq N$ there is only $\xi = 1$ block containing all nodes and the fixed points are the only attractors.

Odd loops with one node delayed by τ^*

Without loss of generality the only inverting Boolean function of the odd loop can be assigned to be in front of the delayed node. As in the synchronous case, there are no fixed point attractors for odd loops. Let again node 1 be updated at time 0. At time T, node 1 will have the opposite state as the original one. After time $2T$, node 1 returns to its original state, which implies that the loop returns to its original state after $2T$ time steps, if it is on an attractor. If all nodes are initially identical, a single domain wall travels around the loop, and after T time steps all nodes are again identical, but with the opposite state.

The shortest attractor has a period $2\tau^*$, and it has alternating blocks. If the number of blocks $\xi = T/\tau^*$ is a prime number, all other attractors have the period $2T$, and the number of different attractors is

$$\nu_\ominus = \frac{2^\xi - 2}{2\xi} + 1 \qquad \text{for prime number of blocks, } \xi \in \mathbb{P}. \tag{4.9}$$

If $\xi \notin \mathbb{P}$, the number of attractors increases as the length of some attractors is shorter. If $\tau^* \geq N$, there is only one attractor with period $2\tau^*$.

Rational delays

It is slightly more general to assume for the nodes to have two different rational delay times. This means there is a single slow node (again node 1) while the other nodes are updated together which can be mapped to the scenario

$$\tau_1 \;=\; \frac{s}{r} \equiv \tau^* \qquad \text{with } r,s \in \{\mathbb{N}_{>0} : s \not| \, r\} \text{ and } \tau_i = 1 \text{ for } i \in \{2,3,\ldots,N\}\,. \tag{4.10}$$

The rational number τ^* is expressed by a fraction in which s is not divisible by r. Node 1 is slower than the others if $s > r$. Using the representation from Eq. (4.10) the condition for the even loop to be frozen can be rewritten as $s \geq N \cdot r$.

4.3.3 Loops with multiple delayed nodes

The next step beyond a loop with one delayed node is a loop with multiple delayed nodes. Let the delay times be integer values, $\tau_i \in \mathbb{N}_{>0}$. The smallest block length τ^* on an attractor is then given by the least common multiple (LCM) of all delays,

$$\tau^* \;=\; \mathtt{LCM}(\tau_i), \tag{4.11}$$

as each node i must produce blocks as output with a length divisible by τ_i. Loops with rational values $\tau_i = r_i/s_i$ can be mapped on those with integer values of τ_i by measuring time in units s of the inverse of the lcm, $s = \mathtt{LCM}s_i$.

A node with delay τ_i "swallows" up to $(\tau_i - 1)$ nodes' values depending on the phase φ_i of the node when a block arrives. The overall number of blocks on the loop depends on the number of swallowed sites, that are between 0 and $(\sum_i \tau_i) - N$ Boolean values. A period T can consist of multiple revolutions around the loop: T has to be a multiple of τ^*.

In the following, two special cases will be considered before focusing on the general case of updating in any order. The section closes by looking at non-integer delays and showing that most loops will be frozen if the delays are real numbers.

Sequential update of the nodes on the loop

By choosing $\tau_i \equiv N \,\forall i$ and initial phases of $\varphi_i = i - 1$ or $\varphi_i = N - i$, a *sequential update* is obtained. Time should be rescaled such that during each time step all N nodes are updated once, a new time step starts as soon as a fixed node j is updated. There are still two possibilities of arranging the phases depending on the order they occur on the loop: The sequential update takes place either connection-wise or counter-connection-wise.

For *connection-wise update* first node i is updated, then $(i+1)$ and so on, i.e., $\varphi_i = i - 1$. After updating each node of an even loop once, all nodes have the same value (as node $i - 1$). Thus, two fixed point attractors exist, namely the two homogeneous configurations with all nodes having the same value, $\boldsymbol{\sigma} \in \{\mathbf{0}, \mathbf{1}\}$. In the case of an odd loop, the attractor has period length $2N$, a single domain wall travels around the loop with one revolution per time step.

For *counter-connection-wise update*, node i is updated before node $(i - 1)$ and $\varphi_i = N - i$. Therefore, N updates give the same result as a single update step in the case of parallel update.

The corresponding results can be adopted, compare Sec. 4.3.2. For that, ξ has to be the number of nodes in the synchronous case, $\xi = N^*$, the results for loops with synchronous update can directly be written down, see Eqns. (4.8) and (4.9).

Loops with same delays but different phases

Another possibility is to choose again $\tau_i \equiv N$ but to update the nodes in any (but fixed) order. In other words, the values of φ_i are some permutation of the numbers 0 to $N-1$.

There are two classes of nodes, according to whether a node is updated before or after its predecessor. Nodes that are updated **after** their preceding node have after N time steps the same state as their predecessor. Such nodes and their predecessor are therefore part of the same *effective node*. The number N^* of effective nodes is identical to the number of nodes that are updated after their predecessor. This also corresponds to the number of nodes a block is able to travel in time τ^*.

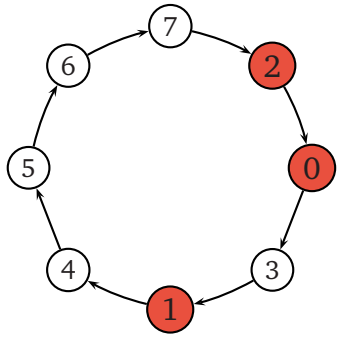


Figure 4.6: Loop with same delay times and different phases $\varphi_i \in \{0,\ldots,N-1\}$. The $N^* = 3$ nodes (red) are updated before their predecessor.

Figure 4.6 shows an example with $N = 8$ nodes, the updating order is given by a tuple $\{\varphi_i\}$ of initial phases in Eq. (4.12). It is possible to specify whether each node is updated before (b) or after (a) its predecessor,

$$\{\varphi_i\} = \{7,2,0,3,1,4,5,6\} \quad \Longrightarrow \quad \{a,b,b,a,b,a,a,a\} \quad \Longrightarrow \quad N^* = 3. \tag{4.12}$$

Once the effective nodes have been identified, N time steps on such a loop with sequential update can be mapped on one time step on a loop of size N^* with parallel update. The number N^* only depends on the permutation of $\{\varphi_i\}$, all results concerning attractor numbers and lengths obtained for loops with synchronous update can then be transfered to loops with sequential update: If N^* is prime and the loop is even, N^* is the length of the attractor measured in units of $\tau^* = \mathtt{LCM}\tau_i \overset{\text{here}}{=} N$. For an odd loop the attractor length doubles.

Different delay times

In general, there are no restrictions for neither the phases φ_i nor the delays τ_i of the nodes except that they are integer values, $\tau_i, \varphi_i \in \mathbb{N}_{>0}$ with $\varphi_i < \tau_i$. In order to determine whether the initial state of a given node $\star$ influences the attractor, the following procedure may be applied. For sake of simplicity, it is described for an even loop here.

1. Some node $\star$ is picked and marked, for instance by fixing the state of this node, $\sigma_\star = 1$. The idea is to evaluate to which nodes this 1 propagates with time. In order to make sure that later on all 1s on the loop will be due to this initial 1, all other nodes' values are set to 0.

2. When the chosen node $\star$ is updated before its subsequent node $\bullet$, the 1 is lost, and the initial state of node $\star$ does not affect the attractor. If the node is updated after its successor, the 1 has moved to the successor and is not yet lost.

3. Next, it is checked whether the successor • is updated before the $\sigma_\bullet = 1$ can propagate further. This check is repeated also for the successors of •, and so on.

During the course of time, the 1 may spread to become a block of larger size, which continues to change its size with time. All the nodes of the block are influenced by the initial condition of node $\star$.

If the loop is now considered at times being a multiple of $\tau^* = \mathtt{LCM}(\tau_i)$, the phases of all nodes are the same as at the beginning. Two different cases for the long term behaviour can occur: After waiting long enough either the original node $\star$ has again state 1, or all 1s are eventually gone. In the first case, the 1 will survive forever, and the initial state of the chosen node $\star$ will consequently affect the attractors. In the second case, the chosen node does not affect the attractors.

The described procedure can be carried out for each node on the loop as starting point. This will identify how many nodes N^* will affect the attractors. From now on, only these *relevant nodes* are taken into account. Moreover, they are only considered at times that are multiples of τ^*.

Let m be the number of relevant nodes through which each block moves during τ^* time steps. The search for the attractors on a loop with any rational delayed nodes is now completed as the problem has been traced back to an already solved one. If m and N^* have no common divisor, the N^* nodes can be ordered in the sequence in which they are visited if the system is only considered at times being multiples of τ^*. The remaining task is to find the attractor number and the length in a loop of size N^*. If m and N^* have a common divisor c, the system can be mapped on c synchronous loops of length N^*/c with parallel update.

Rational and irrational delay times

In nature, it is quite unlikely that latency times have integer values. Without loss of generality $\tau_i > 1$ can be assumed as one can obtain that by rescaling the time. Every rational number τ_i can be expressed by two coprime natural numbers, r_i, s_i which do not have a common divisor

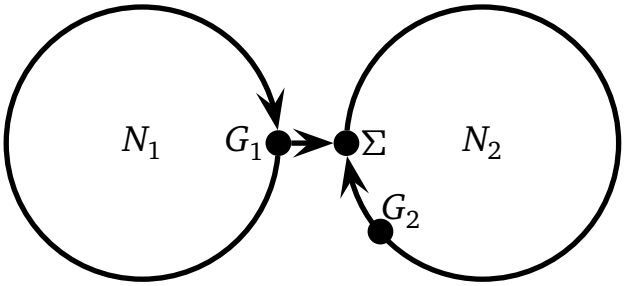
$$\tau_i \;=\; \frac{s_i}{r_i} \text{ with } r_i, s_i \in \{\mathbb{N}_{>0} : s_i \not| \, r_i\}\,. \tag{4.13}$$

The least common multiple of all τ_i gives the size of a block, it is bounded from above, $\tau^* = \mathtt{LCM}\left(\tau_i\right) \leq \prod_i \tau_i$. The loop will be frozen for sure if $s_i \geq r_i N^{1/N}$.

For irrational delay times, the mapping on integer delays cannot be performed. Nevertheless, one can determine whether a value $\sigma_\star$ at a given node $\star$ will eventually form a block that is large enough to never vanish. If all delays have irrational ratios, there will eventually come a moment where all nodes are updated connection-wise, and from then on there is only one block left.

4.3.4 Two loops with a cross-link

For loops, only the two Boolean functions "copy" and "invert" allow non-trivial dynamics. This changes when moving to more complicated relevant components. The ○–○-component is a complex component with exactly one node Σ having two inputs, labelled G_1 and G_2. The component consists of a loop with N_1 nodes connected to a loop with N_2 nodes, see Fig. 4.7.



inputs of Σ		reversible	canalizing functions	
G_1	G_2	$f_\Sigma = f_{\text{rev}}$	$f_{\text{inhomo.}}$	$f_{\text{homo.}}$
0	0	1	1	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

Figure 4.7: The ○—○-component and its Boolean functions at node Σ. This node has two inputs, namely node G_1 and G_2. The labels of the canalizing functions stand for "homogeneous" as $f(0,1) = f(1,0)$ or "inhomogeneous" as $f(0,1) \neq f(1,0)$.

The first loop is either odd or even[2] the second loop can without loss of generality be chosen such that it has only "copy"-functions, except at node Σ. There are no nodes between G_1 and Σ as a system with c nodes on the cross-link between the two loops can be mapped on a system with a direct link. This mapping is performed by connecting node number c of the first loop (counted counterclockwise from G_1) directly with node Σ.

Altogether, there are $2^{2^2} = 16$ possible Boolean functions for a node with 2 inputs. Only the nontrivial cases for the coupling function f_Σ of node Σ are considered now. It should now be shown that three non-trivial functions f_Σ remain which are truly different, they are shown in Fig. 4.7.

In principle, there are twelve *canalizing functions*. For each, a certain value of each input fixes the output, irrespective of the other input. However, only eight of those really depend on two inputs and cannot be reduced to a $k = 1$-function which would separate the two loops. Since the dynamics are invariant under reversing the values of all nodes reversing both the inputs and the outputs of all functions, it is sufficient to consider only four out of these eight functions. Two of the remaining canalizing functions can be neglected because the same dynamics is produced by inverting the values of the nodes of the first loop.

Beside the two frozen functions, $f_0 \equiv 0$ and $f_{15} \equiv 1$, two so-called *reversible functions* are left. If one of the two inputs changes its value the output changes, too. Again, it is sufficient to consider only one reversible function, namely the one which a 0 as output as long as both inputs are not identical, f_9. Reversible functions have *reversible dynamics* for the case without delays; all states are on cycles. The trivial fixed point is $\boldsymbol{\sigma} = \mathbf{1}$.

In the following the results by Kaufman and Drossel [70] for components under synchronous update will be generalised to components with one delayed node.

Delayed node on the first loop of the ○—○-component

If the first loop has a delay τ^*, the value $\sigma(G_1)$ of G_1 can change only at times that are multiples of τ^*. The pattern of change is repeated after the attractor period p_1 of the first loop. If the first loop is on a fixed point, the second loop can be considered as an independent loop with a function at Σ, which depends on the fixed point value of the first loop and on the value of G_2. In other words, the components breaks up into two simple loops.

Therefore, the case that the first loop is not on a fixed point but provides a periodic input of period p_1 to Σ should be considered. There are blocks of size τ^* of identical bits on the first

[2] Once again, an *even loop* refers to a loop with $f_i = \sigma_{i-1}\ \forall i \in \{1,\ldots,N\}$, with $\sigma_0 := \sigma_N$ while for an *odd loop* $f_1 = 1 - \sigma_N$ and $f_i = \sigma_{i-1}\ \forall 1 < i \leq N$.

loop. The second loop then behaves like an independent loop where the Boolean function at Σ changes after τ^* steps, and where the pattern of changes is repeated periodically with period p_1.

If $f_\Sigma = f_{\text{rev}}$, the second loop switches between truth and negation as the effective Boolean function of node Σ, compare Tab. 4.1. For the homogeneous canalizing function f_Σ the second loop changes between "copy" and the constant value $\sigma_\Sigma = 1$ as the effect of node Σ on the second loop. For the inhomogeneous canalizing function, the negation and $\sigma_\Sigma = 1$ alternate. The attractor period is at most $2p_1N_2$ as $2N_2$ is the maximal period length for the second loop.

1^{st} loop	input G_1 of node Σ	$f_\Sigma = f_{\text{inhomogenenous-canal.}}$	$f_\Sigma = f_{\text{homogeneous-canal.}}$
$p_1 = 1$	$\sigma(G_1) \equiv 0$	2^{nd} loop: $\ominus$ (odd)	2^{nd} loop: $\oplus$ (even)
	$\sigma(G_1) \equiv 1$	2^{nd} loop: $\odot$ (frozen)	2^{nd} loop: $\odot$ (frozen)
$p_1 = 2$	$\sigma(G_1) = \{0,1,0,1\ldots\}$	2^{nd} loop: $\ominus,\odot,\ominus,\odot,\ldots$	2^{nd} loop: $\oplus,\odot,\oplus,\odot,\ldots$
$p_1 > 2$	$\sigma(G_1) =$ arbitrary $\{0,1\}$-pattern	2^{nd} loop: $\ominus/\odot$-pattern	2^{nd} loop: $\oplus/\odot$-pattern

Table 4.1: The effect of canalizing rules as function f_Σ of the node with two inputs in the ○—○-component. The second loop can behave as an even loop ($\oplus$), an odd one ($\ominus$) or can be frozen ($\odot$).

More detailed results for possible attractors of such a system under synchronous updating can be found in [70], the only difference being that p_1 is now related in a different way to N_1. An interesting finding is that for a homogeneous canalizing function (delivering zero if and only if both of its inputs are zero), the second loop freezes to the value 1 if p_1 and N_2 have no common divisor.

Furthermore, for an inhomogeneous canalizing function, the first loop enslaves the second loop and imposes its period on it, if p_1 and N_2 have no common divisor. The canalizing blocks can be analysed in detail. Let there be ξ_1 possible blocks in the first loop. Those block candidates are not yet real blocks (as defined in Sec. 4.3.1) because the nodes within a block candidate do not have the same value initially. After some transient time at least some of the candidates become fully filled by 1s. All those blocks of the first loop with value 1 will imprint a 1 in the second loop. The block candidates in between the "on"-blocks will still carry the inverted initial condition during the first revolution around the second loop. Let ξ_2 be the number of possible blocks on the second loop, $\xi_2 = \lceil N_2/\tau^* \rceil$, the ceiling function $\lceil x \rceil$ delivers the smallest integer $z \in \mathbb{Z}$ with $z \geq x$. If ξ_1 and ξ_2 have no common divisor and N_2 is a multiple of τ^*, the states on the attractor can explicitly be pictured: The initial values within the block candidates in between the imprinted "on"–blocks in the second loop cannot be erased, they only toggle every revolution. Altogether, the second loop is enslaved by the first one.

Delayed node on the second loop

In the case where a single delayed node is on the second loop, the first loop behaves in the same way as for simple (synchronously updated) CRBN-loops. Without loss of generality the delay can be assigned to the node Σ. A system with the delayed node m nodes after Σ can be transformed to a system with the delay at Σ by rotating the first loop m nodes counterclockwise.

Node Σ responds to the input from G_1 only every τ^* time steps. The sequence of states at node Σ has the period of the first loop, p_1, if τ^* and p_1 have no common divisor. From now on p_1 should denote the period of the sequence of values of G_1 every τ^* time steps, which is

the period of the input sequence to Σ at those times where Σ is updated. Let furthermore $\xi = \lfloor (N_2 - 1 + \tau^*)/\tau^* \rfloor$ denote the number of blocks of the second loop.

All results for two loops with a cross-link and no delay can now be taken over by replacing N_2 with ξ, by replacing nodes with blocks or block candidates, and by taking τ^* as the time unit. In particular, for a reversible function f_Σ, the largest period is $p_1 \xi \tau^*$. A homogeneous canalizing function f_Σ freezes the second loop on the value 1 if p_1 and ξ have no common divisor. Furthermore, for an inhomogeneous canalizing function, the first loop enslaves the second one and imposes its period, $(\times\tau^*)$ on it if p_1 and ξ have no common divisor.

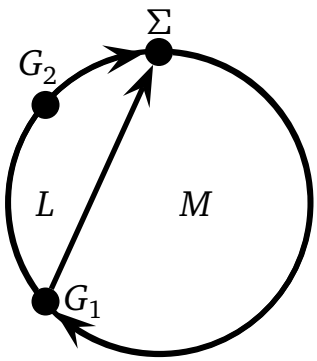
General case

The general case refers to multiple (integer) delays on both loops. Let the first loop have a period p_1 and the second loop p_2 if the latter is even and decoupled from the first one. The general system of two interconnected loops without any delay has been studied in [70]. There, it was shown that possible attractor lengths lie in between p_1 and $2p_1N_2/g$ states on an attractor where $g = \mathtt{LCM}(T_1, N_2)$. These boundaries still hold as introducing delays reduces the number of attractors of the same length by confining the possible patterns on the attractors. Now, the attractor length lies between p_1 and $2p_1p_2/g$ where g is the greatest common divisor of p_1 and p_2, $g = \mathtt{GCD}(p_1, p_2)$.

For a homogeneous canalizing function, the second loop is frozen if p_1 and p_2 are commensurable. The longest attractors occur for reversible functions, $f_\Sigma = f_{\mathrm{rev}}$.

4.3.5 A loop with one additional link

The other complex component having one node with two inputs is a loop with $N = L + M + 2$ nodes and one additional link. The nodes are counted counterclockwise, starting G_2 with the index 1, G_1 is counted as node $L + 1$ and Σ has index N, see Fig. 4.8. The edge $G_2 \to \Sigma$ can be treated as a direct link: A system with $n < L$ nodes in the additional link, $G_2 \to G_1$, an be mapped onto a system with a direct link by connecting node $(L + 1 + n)$ to Σ (if neglecting delays). Furthermore, it is sufficient to look at only five out of the 16 Boolean functions for node Σ, compare Fig. 4.8. The other canalizing or reversible functions yield the same result as they can be mapped onto those five. One only has to invert the output values of the truth table.



$\sigma(G_1)$	$\sigma(G_2)$	$f_\Sigma =$	f_1	f_2	f_{13}	f_{14}	f_9
0	0		1	0	1	0	1
0	1		0	1	0	1	0
1	0		0	0	1	1	0
1	1		0	0	1	1	1

Figure 4.8: The ⊘-component and the Boolean functions for node f_Σ. The functions are named by the decimal representation of the corresponding column of their outputs, for instance $1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 = 13$. The gray shaded entries of the truth table mark the *canalised value* of the function. If the canalised value is fixed, the output of node Σ, f_Σ, is also fixed, otherwise the output depends on both $\sigma(G_1)$ and $\sigma(G_2)$.

The common decimal representation is used for the identifiers: $f_\Sigma \in \{f_1, f_2, f_{13}, f_{14}, f_9\}$. Without loss of generality, the functions of all nodes in the loop are the "copy" function.

Let there be one delayed node with update period $\tau > 1$ in the component. According to the position of this node two cases can be distinguished:

1. The delayed node is among the first $M+1$ nodes (including node G_1). Without loss of generality the delay can be shifted to node Σ.

2. The delayed node is in the chain of nodes between G_1 and Σ and can then be shifted to G_2.

In the first case, the $\oslash$-component can be reduced to a network of effective nodes by looking at the network only every τ^* time steps. Each effective node corresponds to a block of τ^* nodes having the same value. Those blocks are at the same position on the component if $t \cdot \mathrm{mod}(\tau^*) = 0$. The results for the synchronous case (as studied in [70], Sec. 4) hold for the effective variables $\tilde{N}, \tilde{M}$. Again, the smallest integer greater or equal to x is denoted by $\lceil x \rceil := \min_{z \in \mathbb{N}, z \geq x}(x)$.

$$\tilde{N} = \left\lceil \frac{N}{\tau *} \right\rceil \qquad \tilde{M} = \left\lceil \frac{M+2}{\tau^*} - 2 \right\rceil , \tag{4.14}$$

In the following, the second case with node G_2 having a time delay and coupling functions $f_\Sigma \in \{f_{14}, f_{13}, f_1, f_2\}$ will be studied. This corresponds to the choice in [70] where the definition of the functions is not always consistent. f_2 here resembles f_4 in [70], f_{13} is imitates the behaviour of what is called f_{11} in [70].

For each of the five Boolean functions at the node Σ different values for the delay τ and L are investigated. $L = 0$ is just a simple loop as studied above (cf. Sec. 4.3.2). If node Σ is equipped with a canalizing function, the attractors can classified into at least two types, called $\oslash_G$ and $\oslash_M$. In the following, analytical considerations for these types of attractors will be presented and compared with the numerical results.

Attractors of type $\oslash_G$ imposing a condition on $\sigma(G_2)$

Recalling that a canalizing function has the property that one of its inputs alone can determine the output value, two additional terms can be defined, e.g. [68]. The *canalizing value* of a canalizing function is the input value that determines the output value. That output value is called *canalized value*. The Boolean function is said to be on this particular input.

The attractors of type $\oslash_G$ are obtained by requiring that node G_2 never carries its *canalizing value* at the moment when node Σ is updated. The canalizing value is marked in the truth table in Fig. 4.8, the canalizing value is 0 for f_{13} and f_2 and 1 for f_1 and f_{14}.

One part of the $\oslash$ component forms a loop consisting of $M+2$ nodes from Σ to G_1. For the $\oslash_G$-case of G_2 never being at its canalizing value, the loop part of the $\oslash$-component forms an even loop for $f_\Sigma \in \{f_{13}, f_{14}\}$ while for $f_\Sigma \in \{f_1, f_2\}$ the loop part behaves like an odd loop. Just before node Σ is updated, the state $\sigma(G_2)$ and the state of all those nodes, that will in $n \cdot \tau^* \forall n \in \mathbb{N} > 0$ time steps determine $\sigma(G_2)$, must have a value such that $\sigma(\Sigma)$ never is the canalizing value. On all other attractors, G_2 has at least sometimes its canalizing value.

For functions f_{13} and f_{14}, the attractor condition $\oslash_G$ fixes the entire component at the same value if $M+2$ and τ^* have no common divisor, $g := \mathtt{GCD}(M+2, \tau^*) = 1$. If some non-trivial greatest common divisor exists, $g > 1$, only the value of every g^{th} node on the loop part of the

component ($\Sigma \to G_1 \to \Sigma$, $M+2$ nodes) is fixed by this condition. The number of attractors of type $\oslash_G$ is therefore that of an even loop with $(M+2)(g-1)$ nodes and with no delays

$$\nu_{\oslash G}\big|_{f_{13},f_{14}} = \frac{2^{(M+2)(g-1)}-2}{(M+2)(g-1)} + 2\,. \tag{4.15}$$

For functions f_1 and f_2, the condition that G_2 does never have its canalizing value can only be satisfied if τ^*/g is even. The number of attractors of the first type is then that of an odd loop with $(M+2)(g-1)$ nodes and with no delays

$$\nu_{\oslash G}\big|_{f_1,f_2} = \frac{2^{(M+2)(g-1)}-1}{2(M+2)(g-1)} + 1\,. \tag{4.16}$$

Compared to a component with no delays, this new type of attractor increases the mean attractor length if the canalizing function is f_{14}. Without delays, only very short attractors (apart from two fixed points) can occur for f_{14}, see [70]. The increase in attractor length for values $\tau^* > 1$ is clearly visible in the numerical studies exhaustively searching the whole state space, see Fig. 4.9.

Attractors of type $\oslash_M$ where $M+2$ is a multiple of τ^*

An attractor of type $\oslash_M$ is obtained if $M+2$ is a multiple of the delay time τ^*. Again, properties of these attractors can be derived from analytical considerations.

If the loop part with its $M+2$ nodes contains blocks of size τ^*, the dynamics can be mapped on that of an effective component with parameters $\tilde{N}, \tilde{M}$

$$\tilde{N} = \left\lceil \frac{N}{\tau^*} \right\rceil, \qquad \tilde{M} = \frac{M+2}{\tau^*} - 2\,. \tag{4.17}$$

This effective $\oslash$-component does not have any delay but time steps of length τ^*. Again, it is possible to apply the results from the synchronous studies in [70].

In addition to attractors consisting only of homogeneous blocks where all Boolean values of a block are identical, further attractors can be constructed. The idea is that only one bit in each block is the one that triggers node G_2. The value that does not trigger node G_2 does **only** matter when the block reaches G_1: If at this moment Σ is not canalised by G_2, an inhomogeneous block will not be homogenised, but copied or inverted to node Σ. An inhomogeneous block can therefore survive forever if the blocks, that are at G_2 at the moment where the inhomogeneous block is at G_1, do not have their canalizing value. However, this implies that these non-canalizing blocks are copied to Σ from G_1, which is only possible for f_{13}. Indeed, it is known from [70] that a period $\tilde{M}+2$ of attractors on the effective component is only possible for this function, unless $\tilde{N}$ has special values.

Nontrivial attractors of the $\oslash$-component

Non-trivial attractors are those which are found numerically and are neither of type $\oslash_G$ nor $\oslash_M$. They can occur even if $M+2$ and τ^* have no common divisor. For instance, an isolated block of size τ^* of 1s in the initial state will survive forever for the function f_2. This follows as there will be a $\sigma(G_1) = 0$ while the block is at G_2. In fact, there exist a multitude of such non-trivial attractors where there is a 0 at the right position at distance L behind a block of 1s. This explains why the number of attractors found numerically for $\tau^* > 1$ in the exhaustive state space search shown in Fig. 4.9 is larger for f_2 than for the other canalizing functions. The length of these attractors can be larger than $(N+\tau^*-1)$, as can also be seen in Fig. 4.9.

Remarks concerning the four canalizing Boolean functions

The Boolean OR is function f_{14}. It is a so-called homogeneous canalizing function as the function yields the same output for both cases of mixed inputs, $f_{14}(0,1) = f_{14}(1,0)$. The first and the last row in the truth table lead to the two trivial fixed points, $\boldsymbol{\sigma} \in \{\mathbf{0}, \mathbf{1}\}$, as $f_{14}(0,0) = 0$ and $f_{14}(1,1) = 1$.

Cycles in phase space exist if the length of the component's loop part, $(M+2)$, and the block candidate size g is coprime, $g = \mathrm{GCD}(M+2, \tau) > 1$ which is the $\oslash$-condition. In other words: At updating times, the block candidates are always at the same position. Altogether there are $(M+2)/g$ blocks in the loop part. The subsystem reaches the same configuration at maximum after $\mathrm{LCM}(\tau, M+2)$ steps.

Blocks of length τ with mere value 1 can exist in the special case if the blocks fit completely into both the loop part and the whole component

$$g = \tau, \qquad \mathrm{GCD}\left(\left\lceil \frac{L}{\tau} \right\rceil \tau, M+2\right) = n \cdot \tau \qquad \text{with } 1 < n \in \mathbb{N}. \tag{4.18}$$

Unlike the OR-function, f_{13} is an inhomogeneous canalizing function. Mixed input values lead to different outputs, $f_{13}(0,1) \neq f_{13}(1,0)$. There is only one trivial fixed point attractor, $\boldsymbol{\sigma} = \mathbf{1}$ which can be directly seen from the truth table in Fig. 4.8, $f_{13}(1,1) = 1$. Attractors of the component can appear if G_2 is continuously 0 which prevents that the fixed point is reached, $\sigma(G_1) \equiv 0$ which is even more stringent than condition $\oslash_G$.

The Boolean function f_2 is also inhomogeneously canalizing. The statistics of the full enumeration shows that the number ν_τ of attractors for a system with one node delayed by τ is only in rare cases comparable to the synchronous case, $\nu_{\tau=1}$,

$$\nu_\tau = \nu_0^{C \cdot \lfloor N/\tau \rfloor}. \tag{4.19}$$

In there, ν_0 is the number of attractors for a given τ at $N = \tau + 1$ and C is a constant.

The Boolean NOR corresponds to the homogeneous canalizing function f_1. There is no trivial fixed point attractor according to the truth table. Numerically it can be seen that the number of attractors is always smaller than in the synchronous case except for $N = \tau = 4$. For odd τ the number is smaller than for even τ as the number of divisors is smaller. For $\tau = 1$ the maximal attractor length is exactly determined by $\nu_{\max} = 2N$ which is a lower bound for $\tau > 1$.

A reversible Boolean function at node Σ

The last coupling function to be examined is f_9 which is one out of two *reversible functions*. A striking feature of the synchronously updated $\oslash$-component with $f_\Sigma = f_9$ is that cycles of the order of the system size, 2^N, can arise [70].

Not all nodes' values are important in the delayed case, the information of some loops in the L-branch of the component (compare Fig. 4.8) is lost due to the delay at node G_2, such that only $M + 2 + \lceil L/\tau \rceil$ nodes' value are relevant. The exhaustive numerical attractor search suggests that maximal attractor length can be approximated by τ times the maximal attractor length in the non-delayed case for odd τ,

$$A_{\max,\tau} \gtrsim \left|\max\left(A_{i,\tau=1}\right)\right| \cdot \tau\,. \tag{4.20}$$

As in the synchronous case the attractors of reversible functions are longer than in the canalizing case, compare Fig. 4.9.

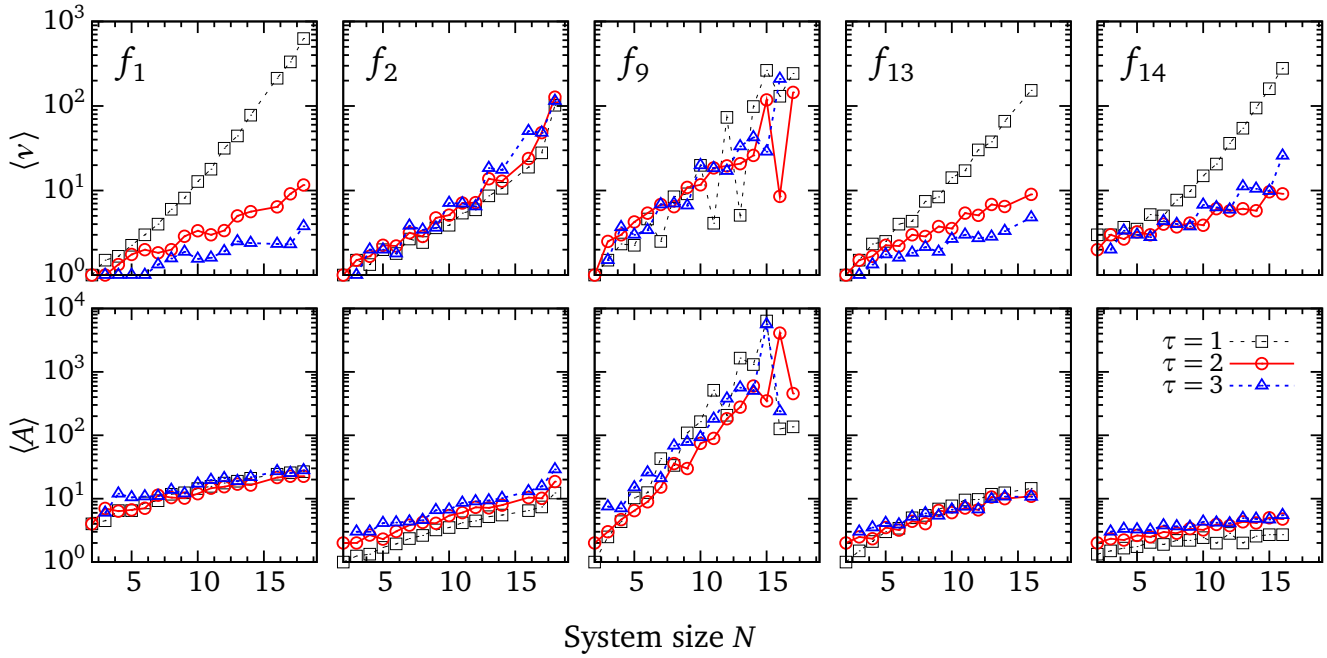


Figure 4.9: Results of the exhaustive state space search for the ⊘-component with small system size, $N < 20$. The upper panel shows the mean attractor number $\langle \nu \rangle$ in dependence of the system size N, the lower shows the mean attractor length $\langle A \rangle$. Both has been carried out for all functions $f_\Sigma \in \{1, 2, 9, 13, 14\}$. In both cases the average was made over all possible realizations which corresponds to an average over all L for a single delayed node with the given delay τ. To keep the diagrams concise only only the smallest delays have been plotted, the simulations included non-trival delay times, $\tau < N - 1$.

4.3.6 Results for the entire network

If all delays are randomly chosen real numbers, loops are most likely to be frozen or on a single attractor. Similarly, more complex components with real delays should have far less attractors than for parallel update.

The attractors of the entire network are obtained from the attractors of the relevant components by the usual considerations [72]: The number of relevant components is of the order of $\ln N$, and usually only the largest component (if any) is more complex than a simple loop. The length of an attractor of the entire network is the least common multiple of the attractor lengths on the components. The attractor number of the network is at least as large as the product of the attractor numbers of the components. The conclusions for the relevant components can therefore by generalised to the entire network.

4.4 Summary

In this chapter, the updating scheme of the N-K-model has been varied. The classical case with parallel update (CRBNs) has been compared with an asynchronous stochastic scheme (ARBNs) and a deterministic asynchronous method (DRBNs). The work was carried out in cooperation with Barbara Drossel, some numerical experiments for the DRBNs have been cross-checked with simulations by Joost Sattler, the results have been published in [57, 59].

It was possible to show for critical ARBNs that the mean number of attractors in a critical Boolean network stochastic update grows like a power law while the mean size of the attractors increases as a stretched exponential with the system size.

Not surprisingly, the delayed nodes in DRBNs typically increase the attractor lengths and reduce the attractor numbers. New types of attractors emerge in the presence of delays. The basins of attraction are naturally larger and thus the path to the attractor becomes more robust.

The biological implications of these results have been pointed out in [78]. Since biological networks do not have a completely synchronous update, the number of attractors should be derived from models with asynchronous update. Some networks, such as in budding yeast [84] appear to be very robust with respect to the introduction of delays. This means that the choice of connections and functions is such that the update sequence does not matter much.

5 Varying the choice of the functions

The simple classification into a frozen and chaotic phase as described in Sec. 2.3.1 is made for networks in which all possible 2^{2^k} Boolean functions occur (k is the fixed in-degree). The mean-field calculation that is usually performed for deciding to which class a network belongs, is based on the assumption that the *proportion of "on"-nodes*, b_t, becomes a constant for long times, $b_t \overset{t\to\infty}{\longrightarrow} b_\infty$. For the proportion of nodes with Boolean value "off" obviously the same assumption holds. However, this does not need to be the case. In fact, this was first seen for a special type of Boolean network where all nodes with the same number of inputs are assigned the same Boolean function [15, 87, 14]. There, the portion of nodes in the "on" state displayed a more complicated temporal behaviour. When the authors varied some parameter, period-doubling cascades and chaos were found in those networks for $b(t)$.

In the present chapter the choice of the ensemble of Boolean functions, $\mathscr{F}$, is varied. A common parameter to select a subset $\mathscr{F}_p$ of all possible functions is the *bias* p, which weights the functions f_i according to their output value [10, 47].

The subset of *threshold functions* abbreviated as $\mathscr{F}_T$, is studied in the current chapter. Details about the implementation are discussed in Sec. 5.1. The aim of choosing $\mathscr{F}_T$ is to build a bridge between classical random Boolean networks with constant b_∞ and Boolean networks with an other ensemble of Boolean functions. Beside the interesting dynamics for b_t, the results may have implications for real networks.

The ensemble of $\mathscr{F}_T$-functions belongs to a rather simple class of networks which allows to investigate its dynamical behaviour in more detail.

Not only the time evolution of the number of "on" nodes b_t is studied (Sec. 5.5), but also the attractors of the networks. First, the criticality conditions are presented (Sec. 5.2), then its simplest version is applied to threshold networks (Sec. 5.4). Later, the focus lays on the parameter range where global oscillations arise and the classical classification breaks down. The presentation follows the one in Ref. [58], where all the results have been published.

5.1 Defining threshold networks

Once again, the basis is the N-K-model which is a directed graph with randomly chosen links between N binary nodes, each of which has K inputs, see Sec. 2.3.1. Throughout the present chapter, the state of a node will be $\sigma_i = \pm 1$, and $+1$ will be called "on" while -1 means "off" (in contrast to $\sigma_i \in \{0,1\}$ in all other chapters). The values σ_i of the nodes are due to historical reasons in the field of threshold networks. This notation is common in literature of neural networks. However, the abundant underlying topology there is usually a weighted graph with real valued nodes, $\sigma_i \in \mathbb{R}$, and sigmoidal[1] coupling functions and layer-topology, see Sec. 2.1.

[1] The classic response of a (artificial) neuron is assumed to be a non-linear but monotonic relation of the dependent variables which shows a characteristic S-shaped curve, referred to as being *sigmoidal*. The Boolean idealisation is a step function.

In the usual formulation of Boolean threshold networks, each edge from input j to node i has assigned a weight c_{ij}. The Boolean function f_i is then defined by a non-linear function (here SIGN) of the sum over all input nodes j of node i,

$$f_i \;=\; \mathtt{SIGN}\left(h+\sum_j \sigma_j c_{ij}\right) \equiv \mathtt{SIGN}\left(s_i\right) \tag{5.1}$$

The *threshold* is defined by h. An *inhibitory* edge has $c_{ij}=-1$ while an *excitatory* has $c_{ij}=+1$. This version of a random Boolean network was for instance used in [107] with $h=0$ or for the real-world example of baker's yeast presented in App. B. In agreement with other authors, it is understood that $\mathtt{SIGN}(0):=1$.

The ensemble $\mathscr{F}_T$ of threshold functions

General threshold functions can be expressed as a truth table. Altogether, there are 2^k input configurations to be specified for $f_i\left(\sigma_{i_1},\ldots,\sigma_{i_k}\right)$. As the order of the inputs is insignificant for threshold functions, it is sufficient to regard only $(k+1)$ different input configurations. For in-degree k there are k different possible threshold ranges yielding different results for f_i, their mappings to a truth table are listed in Tab. 5.1.

As defined in Sec. 4.3.4, a *canalizing function* has the property that at least one of the input variables is able to determine the output regardless of the other value. For $k=2$ two possible threshold ranges exist, $-2\le h<0$ and $0\le h<2$, ignoring the trivial case of constant Boolean functions which occurs for $h=k$. Both resulting functions are *canalizing*.

Other studies focusing on the ensemble of canalyzing functions only have been motivated by the finding that gene regulation networks appear to have many canalyzing functions [102, 68].

Table 5.1: Mapping of threshold networks to Boolean truth tables for different in-degrees k. The first block of columns gives the $k+1$ different input configurations, the order of the inputs does not matter. Then, the sum $\Sigma=\sum_j \sigma_j c_{ij}$ for $c_{ij}\equiv 1$ over all inputs is specified. The other columns are entitled by the interval for the threshold h and show the corresponding output values. The gray scale is to clarify the values ± 1.

$k=4$				$\sum_{j=1}^{k}\sigma_j c_{ij}$	$h\in[-4,-2)$	$h\in[-2,0)$	$h\in[0,2)$	$h\in[2,4)$
−1	−1	−1	−1	−4	−1	−1	−1	−1
−1	−1	−1	+1	−2	−1	−1	−1	+1
−1	−1	+1	+1	±0	−1	−1	+1	+1
−1	+1	+1	+1	+2	−1	+1	+1	+1
+1	+1	+1	+1	+4	+1	+1	+1	+1

$k=3$			Σ	[-3,-1)	[-1,1)	[1,3)
−1	−1	−1	−3	−1	−1	−1
−1	−1	+1	−1	−1	−1	+1
−1	+1	+1	+1	−1	+1	+1
+1	+1	+1	+3	+1	+1	+1

$k=2$		Σ	$[-2,0)$	$[0,2)$
−1	−1	−2	−1	−1
−1	+1	±0	−1	+1
+1	+1	+2	+1	+1

The ensemble $\mathscr{F}_T$ of functions to be considered in the present chapter is tuned by a parameter p_+. A link is excitatory ($c_{ij} = +1$) with probability p_+ and inhibitory ($c_{ij} = -1$) with probability $(1 - p_+)$. The value of the threshold is set to $h = 0$ and the in-degree to $k = 2$. Since each of the two input connections can be excitatory or inhibitory, the model has four different update functions, which are listed in Tab. 5.2 together with their weights and which is more general than the threshold functions shown in Tab. 5.1. Beside that, those functions are a subset of the ones classified as *biologically meaningful* by Luc Raeymakers [105].

5.2 Different criticality conditions for the dynamics

As just defined, the ensemble is $\mathscr{F}_T = \{f_7, f_{11}, f_{13}, f_{14}\}$. For all these update functions the probability is 0.5 that the output changes if one input value is flipped. Since each node is on average the input to two other nodes, a perturbation at one node propagates on average to one other node. Therefore, one should expect the model to be critical according to the classical criterion. This is in agreement with the finding that $k = 2$-RBNs containing only canalizing update functions (but all of them with the same probability) are critical [102]. However, the simple argument is based on the assumption that all four possible input configurations occur equally often, which may be true at the beginning of a simulation run, but may be wrong already after one time step.

Next, a concise overview of the most important methods to determine criticality is given and later on applied to threshold networks.

Saturation method for criticality

The approach to look at the fixed point of the *overlap* $x(t)$ between two different network states is termed *saturation method*. It is closely related to the *Hamming distance* $H(t)$ between two different initial conditions of a network realization, $\{\sigma_i^{(1)}, \sigma_i^{(2)}\}_{i=1\ldots N}$, as mentioned in Sec. 2.3.1. In short, the *overlap*

$$x(t) \;=\; 1 - \frac{H(t)}{N} \equiv 1 - \frac{1}{4N}\sum_{i=1}^{N}\left(\sigma_i^{(1)}(t) - \sigma_i^{(2)}(t)\right)^2 \qquad \text{for } \sigma_i \in \{\pm 1\} \tag{5.2}$$

is the fraction of nodes having the same value in both copies of the network. It varies with time, $x(t) \in (0,1)$, and approaches $x \approx 0.5$ if the two configurations are totally independent.

The method assumes that the overlap becomes constant ("saturates") after long times, $x_\infty = x(t \to \infty)$, compare dark gray area in Fig. 5.1. This value is considered as an order parameter of the system, distinguishing whether initial perturbations will die out or propagate across the entire system. The system is said to be emphcritical if

$$\left.\frac{\mathrm{d}F(x(t))}{\mathrm{d}x(t)}\right|_{x(t)\to 1} \;=\; 1 \qquad \text{where } F(x(t)) \equiv x(t+1)\,. \tag{5.3}$$

All that is performed assuming that the two states of a network are virtually the same after long times.

The *saturation method* is well-known as Bernard Derrida's *annealed approximation* [45, 44], a mean-field theory ignoring possible correlations between nodes. The network is assumed to be infinitely large and it is neglected that the network realizations are *quenched* (in the usual sense,

that the function f_i do not change in time). The approximation means that the behaviour of the network dynamics remains the same when randomly reassigning a new realization at each time step. Then, it is possible to write the time evolution of the overlap for the ensemble $\mathscr{F}_p$ as

$$x(t+1) \;=\; x(t)^k \cdot 1 + \left(1 - x(t)^k\right) \cdot \left(p^2 + (1-p)^2\right). \tag{5.4}$$

With probability $x(t)^k$, all k input values are the same, the two configurations will overlap. With probability $(1 - x(t)^k)$ not all k input node values are the same, only in that case the overlap can change in the next time step. However, it is possible that the two outputs are the same, which is given by the two summands depending on the common *bias* p. The criticality condition Eq. (5.3) leads to the expression $2p(1-p)k_c = 1$ with critical connectivity k_c.

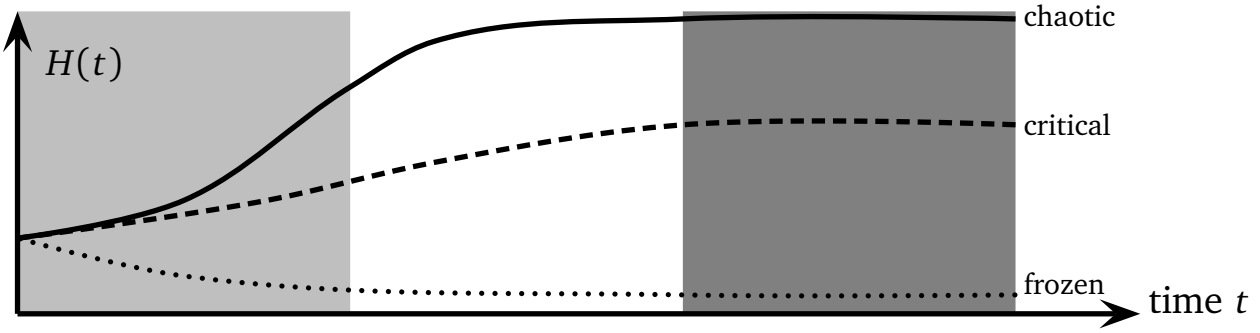


Figure 5.1: Sketch of the Hamming distance's time evolution, $H(t)$, of chaotic, critical and frozen network realisations. The perturbation method Eq. (5.5) focuses on the left (light gray) area while the saturation method looks at the right (dark gray) area.

Perturbation method

Among others, Maximo Aldana studied the percolation of information [10] (pp. 11f). Again, the starting point are two random initial states such that the majority of the nodes have the same value in both configurations, but still they differ in a large number of nodes' values, namely $H_0 = H(t=0) \neq 0$.

Again, the method should be explained by looking at the example of a classical N-k-model network with a function ensemble $\mathscr{F}_p$ tuned by the *bias* p. Changing a single node's value will on average change the arguments of k functions. Overall, there will be $k \cdot H(0)$ functions affected. Each function will change its outcome with bias p as the functions are (quite) randomly distributed in the network. This does only hold as long as p is the same as the probability that an arbitrary node has value 1 in the stationary state, b_t.

Assuming that the functions in next time step still can be chosen at random, i.e., that $k \cdot H(t)$ nodes are affected, this leads to

$$H(t+1) = p \cdot k \cdot H(t) \Longrightarrow H(t) = H_0 \exp\left(\ln(p \cdot k) \cdot t\right). \tag{5.5}$$

Again, the critical case is $k_c = 1/p$ for the ensemble $\mathscr{F}_p$. For $k > k_c$, the number of changed elements grows exponentially with time, while for $k < k_c$ it decays exponentially, see left part of Fig. 5.1. As state space is finite, Eq. (5.5) is valid only for small times.

Generalised method

In general, the *bias* p is not the same as the probability b_t that an arbitrary node has the value 1 in the stationary state, $p \neq b_t$. André Moreira and Luís A. Nunes Amaral argued [97] that the calculations should be performed such that the input configurations are weighted with their frequencies in the *stationary state*. The dark gray shaded area in Fig. 5.1 illustrates this stationary state. This type of calculation is explained in great detail in [74]. A similar type of calculation was applied earlier to a lattice model, where the approximation is not exact [126].

The generalised method aims to determine the critical behaviour of Boolean systems built from an arbitrary ensemble $\mathscr{F}$ of Boolean functions. André Moreira and coworkers define the *influence of a function's* j^{th} *input*, $I_j^{(1)}(f_i)$, as the probability that the function f_i changes its value when the j^{th} input changes. The *average influence of a function* is given by

$$I^{(1)}(f_i) \equiv \frac{1}{k_i} \sum_{j=1} k_i I_j(f_i). \tag{5.6}$$

More general, $I^{(k_d)}$ is the influence of k_d input variables. The probability that an arbitrary node's value is flipped in the next time step depends on the number k_f of flipped inputs and on their influence $I^{(k_f)}$. With a binomial distribution of k_f one obtains a condition for criticality only depending on the average influence of one variable in the ensemble $\mathscr{F}$,

$$I^{(1)}\left(\{\mathscr{F}\}, k_c\right) \cdot k_c \;=\; 1 \qquad \text{where } I(\mathscr{F}) \equiv \langle I^{(1)}\left(f_i\right)\rangle_{\mathscr{F}}\,. \tag{5.7}$$

The difficulty in using this equation lays in computing the influence of the variables of the Boolean functions. The required information cannot just be compiled by using the truth table consisting of the outcome of the functions of $\{\mathscr{F}\}$. For example the functions $\mathscr{F}_T = \{f_7, f_{11}, f_{13}, f_{14}\}$, see Tab. 5.2, have bias $p = 0.75$ each, but f_7 behaves completely different than the others (it is the only function without any trivial fixed point of the form $f(\sigma_1, \sigma_2) \equiv \pm 1$).

5.3 Probability to flip between the two outcomes

The influence on the output of a general threshold function is evaluated for a given number a of changed input values. In the following derivation, the parameter π_{k_d} will be equivalent to the influence of k_d input variables, $\pi_{k_d} \equiv I^{(k_d)}$. This will lead to the probability π_a that the output changes, given a certain number of flipped inputs. The definition already appearing in Eq. (5.1), $s_i := \sum_j \sigma_j c_{ij}$, will again be used (note that $h = 0$).

Table 5.2: The four possible threshold functions for $k = 2$. The input configuration, the functions with their name, outcomes and probabilities are specified.

input configuration		f_7	f_{11}	f_{13}	f_{14}
−1	−1	+1	+1	+1	−1
−1	+1	+1	+1	−1	+1
+1	−1	+1	−1	+1	+1
+1	+1	−1	+1	+1	+1
probability of the function		$(1-p_+)^2$	$p_+(1-p_+)$	$p_+(1-p_+)$	p_+^2

Changing exactly one input

Let W be the probability that s_i does not change its sign, i.e., it stays on the side of the threshold where it was before flipping one input. The number of successful events (with no change in the output) is the cardinality of the set for which $s_i \notin [-1,+1]$ holds. The sum s_i can vary between $-k$ and $+k$ as can also be seen in Tab. 5.1. The outcome is fixed by the number z of summands in s_i being positive, i.e. how many $\sigma_j c_{ij} = +1$.

Thus, $s_i = -k + 2z$ before the update, the value after the update is $s_i' = -k + 2z'$. If $z = k/2$ (even number of inputs), the output changes if $z' < k/2$. For odd number of inputs the switching occurs if $z = (k-1)/2$ and $z' = (k+1)/2$ or vice versa.

Putting all that together, the complement of the desired probability, $\overline{W}$ can be written down for $s_i \in [-1,+1)$,

$$W \;=\; 1 - \overline{W} = 1 - \frac{g}{\sum_{z=0}^{k} \binom{k}{z}} = 1 - \frac{d}{2^k}\,, \tag{5.8}$$

this is the probability that a node remains unchanged given that exactly one input changes. Here, for g two cases have to be distinguished depending on whether k is even or odd as this affects the position of the threshold. A factor of $1/2$ accounts for the fact that s_i can change in two directions of which only one is towards the threshold.

$$g^k_{\text{even}} = \frac{1}{2}\begin{pmatrix} k \\ \frac{k}{2} \end{pmatrix} \qquad 2 \cdot g^k_{\text{odd}} = \begin{pmatrix} k \\ \frac{k-1}{2} \end{pmatrix} + \begin{pmatrix} k \\ \frac{k+1}{2} \end{pmatrix} = \begin{pmatrix} k+1 \\ \frac{k+1}{2} \end{pmatrix} = 2 \cdot g^{k+1}_{\text{even}} \tag{5.9}$$

Using JAMES STIRLING's approximation, $n! \approx e^n n^n$, for the limiting case of large number of inputs, k, one sees that $\overline{W} \sim 1/\sqrt{k}$. This makes sense and will also hold for continuous weights c_{ij}: The central limit theorem predicts a Gaussian distribution of width $\sqrt{k}$. If the width of the distribution grows like $\sqrt{k}$, the probability $\overline{W}$ to be in an interval in the order of 1 around the threshold decreases with $1/\sqrt{k}$.

Changing more inputs

Now, an arbitrary number a of changing inputs is assumed and the corresponding probability π_a for a changing output s_i will be evaluated. Again, the sum s_i is determined by z, the number of effective inputs being positive. The threshold is excessed if z is below the threshold, $z \in ((k-a)/2, a/2]$, while after the flipping of a inputs it is above, $z' \in [a/2, (k+a)/2]$.

Furthermore, there is the probability $p(z)$ to start at a certain z before flipping a nodes. The probability $p(z'|z)$ to reach a certain z' given a starting value z is more difficult to express,

$$p(z) = \binom{k}{z} 2^{-k} \qquad p(z'|z) = \sum_{\{\eta\}} \prod_{i=1}^{a} \begin{Bmatrix} \frac{z_i}{k} \\ 1 - \frac{z_i}{k} \end{Bmatrix} \qquad \text{if} \qquad \eta_i = \begin{cases} -1 \\ +1 \end{cases}. \tag{5.10}$$

The sum is taken over all permutations, $\{\eta\} = \{\eta_1, \ldots, \eta_a\}$, of a sequentially performed flipping events which specify a path of reaching z' starting from a given z. For each flipping event η_i, a single input node changes its value, either from $-1 \to +1$ or $+1 \to -1$, only the first type moves s_i towards the threshold as z is below. The inner product in Eq. (5.10) is the probability of taking a certain *path of switching events* $\{\eta\}$,

$$\{\eta\} \;=\; \{(z \equiv z_0) \to \cdots \to (z_i = z_{i-1} + \eta_i) \to \cdots \to (z_a \equiv z')\}\,. \tag{5.11}$$

Each term of the product is the probability to move flip an input in a given step of the path.

The overall probability π_a that a change of the output s_i occurs given a number a of inputs changes is obtained by summing over all starting values z being on one side of the threshold and over all z' being on the other side

$$\pi_a = \sum_{z\in\left(\frac{k-a}{2},\frac{a}{2}\right],z'\in\left[\frac{a}{2},\frac{k+a}{2}\right]} p(z)\cdot p(z'|z) + \sum_{z\in\left[\frac{a}{2},\frac{k+a}{2}\right],z'\in\left(\frac{k-a}{2},\frac{k+a}{2}\right]} p(z)\cdot p(z'|z). \tag{5.12}$$

If k is odd, for symmetry reasons the first sum is doubled. The above equation describes in mathematical terms the algorithm how to calculate the switching probabilities for any general a and k.

For $a = 1$ it is easy to see that the algorithm works: Without loss of generality let $g = g_{\text{even}}$. As there is only one pair (z, z') allowed, $\{\eta\} = \{+1\}$ holds, leading to

$$\pi_{a=1} = \sum_{z=\frac{k}{2},z'=\frac{k}{2}+1} \underbrace{\binom{k}{\frac{k}{2}}\frac{1}{2^k}}_{p(z)} \sum_{\{\eta\}}\prod_{i=1}^{a=1}\left(1-\frac{\frac{k}{2}}{k}\right) = \frac{1}{2^{k+1}}\cdot\binom{k}{\frac{k}{2}}. \tag{5.13}$$

This is the intuitive results that each of the (equally probable) input 2^{k+1} input configurations, exactly $k/2$ out of k have to be above the threshold in order to guarantee that the output switches if one input is flipped. In other words, z has to lay exactly below the threshold and z' exactly above (or vice versa).

5.4 Application to threshold networks

From now on, the focus again lays on $k = 2$ networks with the parameter p_+ as defined in Sec. 5.1. The generalised method for criticality is applied to determine for what values of p_+ it predicts that the model is frozen, critical, or chaotic. In the thermodynamic limit the proportion of "on"-nodes, b_t, changes deterministically according to

$$b_{t+1} = 1 - \Big[\underbrace{b_t^2\left(1-p_+\right)^2}_{\text{for } f_7} + \underbrace{\left(1-b_t\right)^2 p_+^2}_{\text{for } f_{14}} + \underbrace{2\cdot b_t\left(1-b_t\right)p_+\left(1-p_+\right)}_{\text{for } f_{11}\wedge f_{13}}\Big]. \tag{5.14}$$

The expression in the square brackets is the probability that an input combination leads to $\sum_j c_{ij}\sigma_j = -2$ yielding an output -1. In the stationary state holds $b_{t+1} = b_t = b$ with

$$b(p_+) = \frac{4p_+^2 - 2p_+ - 1 \pm \sqrt{5 - 12p_+ + 8p_+^2}}{2(1-2p_+)^2}. \tag{5.15}$$

The sign $\pm$ in the numerator has to be chosen such that $b \in [0,1]$, therefore only the positive branch remains, see Fig. 5.2 where Eq. (5.15) is plotted.

For $p_+ = 1/2$ all four Boolean functions occur with the same probability. As the output in the truth-table is $+1$ in three quarters of the cases, one expects $b = 3/4$ which coincides with the result obtained from Eq. (5.14) as the stationary solution.

For $p_+ = 1$ there are only positive link weights, $c_{ij} \equiv +1$, and the only Boolean function which appears in the network is f_{14}. Fixed points are obviously the two homogeneous cases with all nodes having the same value, $f_{14}(-1,-1) = -1$ and $f_{14}(+1,+1) = +1$. For the stationary value of b_t, it is $b^* = 0$ and $b^* = 1$, with the first solution being unstable as it is destroyed by a single node in the state $+1$. The second solution is a stable fixed point of the dynamics.

For $p_+ = 0$, two inputs -1 lead to an output of $+1$ as only the function f_7 is present, it is the *golden ratio*[2] $b = (-1+\sqrt{5})/2$.

Probability π_1 for a change when flipping one input

Let π_1 be the probability that the output of a node changes in the stationary state given that exactly one input is inverted. The mean number of nodes to which a perturbation at one node propagates is in the stationary state given by $2\pi_1$. This can be evaluated by adding together the probabilities for those input configurations which allow a transition from an output of $+1$ to -1 and vice versa. This transition is possible for half of the input configurations leading to $s_i = \sum_j \sigma_j c_{ij} = 0$ (the first 4 terms in the following equation) and for all input configurations for which $s_i = -2$ (the last 4 terms),

$$\begin{aligned}\pi_1 = \Bigg\{ & \begin{array}{ll} (1-p_+)(1-b)(1-p_+)b & +p_+ b p_+(1-b) \\ + \; p_+ b(1-p_+)b & +(1-p_+)(1-b)p_+(1-b) \end{array} \Bigg\} \text{ for } \textstyle\sum_{j=1}^{k} \sigma_j c_{ij} = 0 \\ & \begin{array}{ll} (1-p_+)b(1-p_+)b & +p_+ b(1-p_+)(1-b)+ \\ + \; (1-p_+)(1-b)p_+ b & +p_+(1-b)p_+(1-b) \end{array} \Bigg\} \text{ for } \textstyle\sum_j \sigma_{j=1}^{k} c_{ij} = -2 \\ & = b + p_+ - 2bp_+ \,. \end{aligned} \tag{5.16}$$

In this ensemble of networks, only those canalizing functions occur, which truly depend on two inputs and one input is canalizing. For this case it is known that $\pi_1 = 1/2$, because the canalizing input (which actually changes something) is picked in half of the cases. For example, in f_7 an input configuration $(+1,-1)$ only leads to a modification of the output if one flips the second input, $f_7(+1,-1) = f_7(-1,-1) \neq f_7(+1,+1)$. Equation (5.16) yields the same result as the this reasoning, $\pi_1 = 1/2$.

Setting $p_+ = 1$ leads to $\pi_1 = 1 - b$. In the stationary state the system is most probably on the (stable) fixed point $b = 1$. Flipping an input from $+1$ to -1 never changes the output in function f_{14}, therefore $\pi_1 = 0$.

For the case $p_+ = 0$ (function f_7 only) there is no simple illustration. The stationary value according to Eq. (5.16) is given by the *golden ratio*, $\pi_1 = b \approx 0.618$.

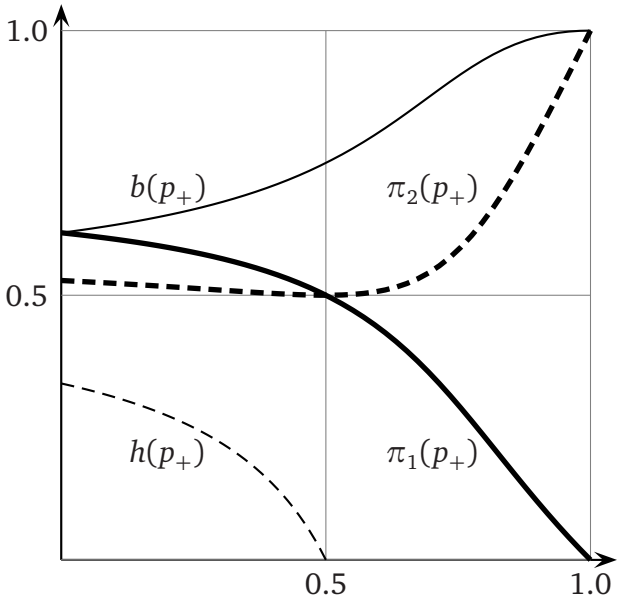


Figure 5.2: The functions $b(p_+)$ as given by Eq. (5.15), $\pi_1(p_+)$, see Eq. (5.16), $\pi_2(p_+)$, see Eq. (5.17) and the stationary value $h(p_+)$ of Eq. (5.18) are plotted as a function of the probability for an activating link, p_+.

[2] The *golden ratio* is a value φ for which holds $\frac{a+b}{b} = \frac{a}{b} \equiv \varphi$.

In a nutshell, the model is in the frozen phase for $p_+ > 1/2$, it is critical for $p_+ = 1/2$ and chaotic for $p_+ < 1/2$.

Probability for a change when flipping two inputs

To obtain the *probability for a change when flipping two inputs*, π_2, all input combinations leading to $\sum_j \sigma_j c_{ij} \in \{\pm 2\}$ are summed up. There are altogether eight terms which cannot be simplified much

$$\begin{aligned}
\pi_2 &= \left\{ \begin{array}{lll} & p_+ b p_+ b & +(1-p_+)(1-b)p_+ b+ \\ + & p_+ n(1-p_+)(1-b) & +(1-p_+)(1-b)(1-p_+)(1-b) \end{array} \right\} \text{ for } \textstyle\sum_{j=1}^{k} \sigma_j c_{ij} = +2 \\
&\quad \left\{ \begin{array}{lll} & (1-p_+)b(1-p_+)b & +p_+ b(1-p_+)(1-b)+ \\ + & (1-p_+)(1-b)p_+ b & +p_+(1-b)p_+(1-b) \end{array} \right\} \text{ for } \textstyle\sum_{j=1}^{k} \sigma_j c_{ij} = -2 \\
&= 1 - 2b(1-2p_+)^2 + 2b^2(1-2p_+)^2 - 2p_+ + 2p_+^2 \,. \qquad (5.17)
\end{aligned}$$

The behaviour of $\pi_2(p_+)$ is shown in Fig. 5.2. One can see that $p_+ = 1/2$ means $\pi_2 = 1/2$ which can be understood intuitively: For a pure $\mathscr{C}_2$-network it does not matter how many inputs are flipped, the probability that the outcome changes is always the same.

Both $p_+ = 0$ and $p_+ = 1$ simplify Eq. (5.17) to $\pi_2 = 1 - 2b + 2b^2$. Only one Boolean function occurs, f_7 for $p_+ = 0$ and f_{14} for $p_+ = 1$. For the fixed points of those functions holds $b \in \{0, 1\}$, which means that a change of the output is the certain event, i.e., this event has probability 1, $\pi_2 = 1$. The explanation is that for $k = 2$ inputs, a modification of 2 inputs is only possible if the input values are in a homogeneous configuration.

Stationary value of the overlap

For the case that each node has exactly $k = 2$ inputs, the probability that the output changes in the next step of the system's time evolution is a sum of two influences. On the one hand, there is the probability that exactly one input changes times the probability π_1 that such a change does also change the output. On the other hand, the probability for two changing inputs multiplied by the probability that changing two inputs does really affect the output of the function needs to be taken in account. The latter case involves π_2. Thus, the time evolution for the normalised Hamming distance h_t is:

$$h_{t+1} = 2h_t(1-h_t) \cdot \pi_1 + h_t^2 \cdot \pi_2 \,. \qquad (5.18)$$

By using $h_t = 1 - x_t$, the criticality condition for the saturation method substantiates to

$$\begin{aligned}
\left. \frac{\mathrm{d}x_{t+1}}{\mathrm{d}x_t} \right|_{x \to 1} &= \left[\frac{\mathrm{d}}{\mathrm{d}x} \Big(1 - (2x(1-x)\pi_1 + (1-x)^2 \pi_2) \Big) \right]_{x \to 1} \qquad (5.19) \\
&= \big[-(2-4x)\pi_1 - \underbrace{(-2+2x)}_{\longrightarrow 0 \text{ for } x \to 1} \pi_2) \big]_{x \to 1} = 2\pi_1 \qquad (5.20)
\end{aligned}$$

The second part with π_2 drops out and the result obtained by using the saturation method is the same as the one of the perturbation method, $2\pi_1 = 1$. This observation can be understood in a more general way. The probabilities π_a of higher order changes $a > 1$ are negligibly small for critical networks.

Implication

If h_t is very small, in the time evolution of the normalised Hamming distance, Eq. (5.18), terms proportional to h_t^2 can be neglected. This leads to the approximation

$$h_{t+1} \simeq 2 \cdot h_t \cdot \pi_1 , \tag{5.21}$$

which allows for the growth of a small perturbation if $2\pi_1 > 1$. This means in agreement with the results above that $p_+ < 1/2$, compare the thick solid line in Fig. 5.2. The transition from a stationary value $h = 0$ to a stationary value $h > 0$ occurs at the same point (thin dashed line in Fig. 5.2).

In Figs. 5.4 and 5.5, for a given initial condition of the network fixed by b_0, the time evolution of the Hamming distance h_t is given for different values of p_+.

In in Fig. 5.4 it is possible to distinguish the classical three phases. If the network is frozen, h_t will vanish as for $p_+ = 0.6$ (blue dashed-dotted line). In the chaotic phase, h_t will remain constant, $h_\infty = \text{const.} > 0$, for instance for $p_+ = 0.4$ (black solid line). The transition between those two phases occurs for $p_+ = 0.5$ (red dashed line). For $p_+ = 0.11$ the Hamming distance still converges to a constant, although there is some oscillation present (black lines with points). In Fig. 5.5 the oscillations of period 2 can be seen even clearer. They are an indication that the dynamics in the *chaotic* phase has some structure, which is studied in the following.

5.5 The non-frozen regime

The chaotic phase in now further investigated based on the evaluation of π_1 at the fixed point value of the fraction of "on"-nodes, b^*. In fact, the dynamics is not chaotic at all for sufficiently small p_+. The reason is that the calculations were based on the assumption that b becomes stationary for large times. In order to see that this need not be the case, a good starting point is the situation where $p_+ = 0$: Then, the time evolution in Eq. (5.14) writes $b_{t+1} = 1 - b_t^2$. This is a one-dimensional map with a fixed point b^*, see Fig. 5.3. The fixed point b^* is the intersection of the first bisecting line (dashed) with b_{t+1} (solid).

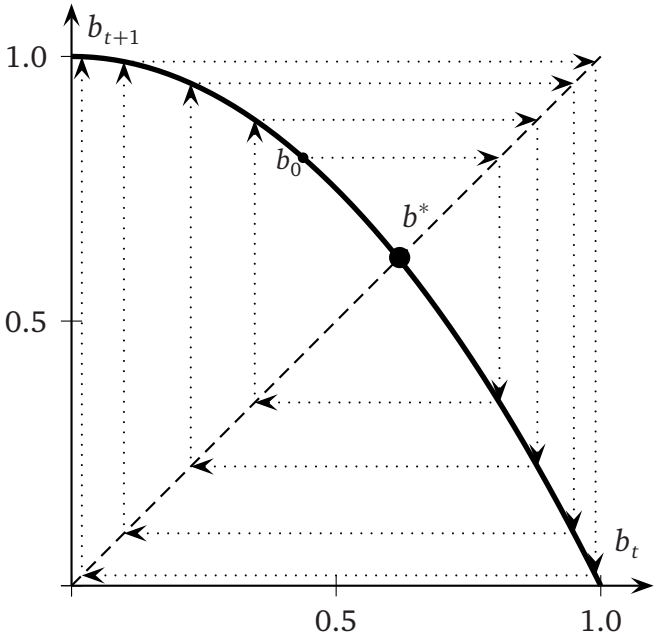


Figure 5.3: The map b_{t+1} vs. b_t for $p_+ = 0$. The fixed point $b^\star$ is unstable. The dotted arrows depict a sample trajectory starting from $b_0 = 0.4375$ which eventually leads to an attractor of length 2.

The fixed point is unstable, just as is found in specific random Boolean networks where all nodes are assigned the same function [15]. Instead of having a stationary point with a constant proportion of nodes in the one of the two states, the system oscillates between a configuration where all nodes are switched on and a state where all nodes are switched off. This is not chaotic dynamics at all but very stable dynamics. In order to determine the range of p_+ values, for which the fixed point value of b^* is unstable, a linear stability analysis is

performed. The ansatz $b_{t+1}(b^* + \delta b_t) = b^* + \delta b_{t+1}$ leads in linear order in δb and for $p_+ < 1/2$ to

$$\delta b_{t+1} = -2\left(b_t\left(1-2p_+\right)^2 + \left(1-2p_+\right)p_+\right)\delta b_t \tag{5.22}$$

$$= \left(1 - \sqrt{5 - 12p_+ + 8p_+^2}\right)\delta b_t =: M \cdot \delta b_t \tag{5.23}$$

To obtain this, Eq. (5.15) was used. The fixed point of a discrete system is stable as long as the real parts of the eigenvalues are less than unity, thus b^* is stable if the real part of M is smaller than 1, which is the case if

$$p_+ \quad > \quad \frac{3-\sqrt{7}}{4} \equiv p_{\mathrm{cb}} \approx 0.0886\,. \tag{5.24}$$

The system has a stationary state with constant proportions of nodes being “on” and “off” at the critical value p_{cb}.

5.5.1 Oscillations with period 2

Now, the region $p_+ < p_{\mathrm{cb}}$ will finally be investigated in more detail. In this interval, the proportion of “on” and “off” nodes oscillates. For $p_+ = 0$, every node oscillates with period 2, and the global attractor has period 2. However, this need not necessarily be the case if b oscillates with period 2. The attractor could be much larger, while the proportion of off and on nodes oscillates still with period 2.

In order to determine for which parameters an attractor with period 2 is stable, again linear stability analysis is performed but now for every second time step. The assumption is that the system is on an attractor of length 2. Let there be at every even time step a proportion x of “on”-nodes and every odd step a proportion y.

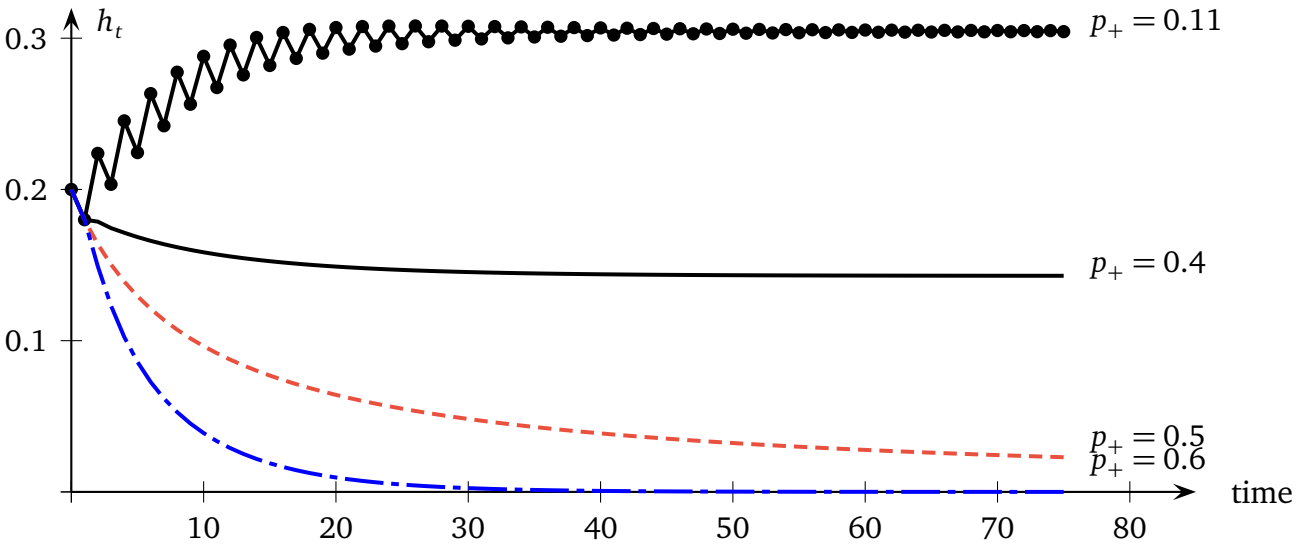


Figure 5.4: Time evolution of the Hamming distance h_t for $p_+ > 0.1$. The time evolution according to Eq. (5.14) is shown for the initial condition $h_0 = 0.2$ and $b_0 = 0.5$.

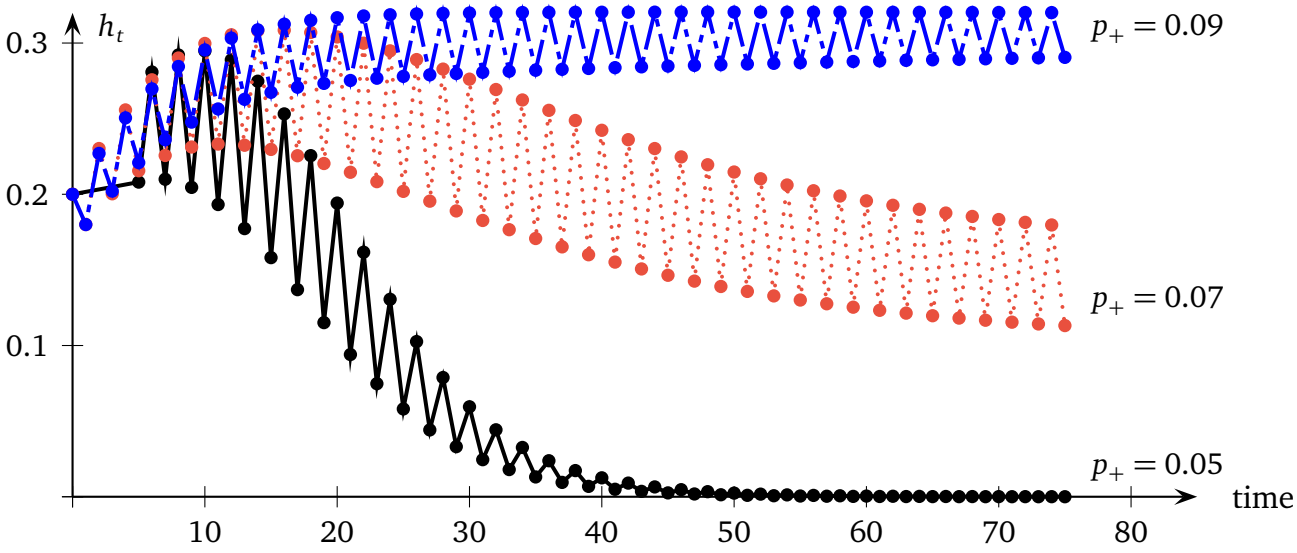


Figure 5.5: Time evolution of the Hamming distance h_t for $p_+ < 0.1$. Again, the curves are calculated according to Eq. (5.14) starting from $h_0 = 0.2$ and $b_0 = 0.5$. For smaller values of p_+ oscillations with period 2 occur, see text for details.

One node is flipped and it is evaluate how the Hamming distance grows in comparison to the undisturbed system after two time steps. The condition that a perturbation of one node propagates on average to one other node after two time steps is equivalent to

$$\pi_1(x) \cdot \pi_1(y) \;=\; \frac{1}{4} \tag{5.25}$$

Combining Eq. (5.25) with the time evolution of x and y as given by Eq. (5.14), three equations,

$$y = 1 - [x^2(1-p_+)^2 + (1-x)^2 p_+^2 + 2x(1-x)p_+(1-p_+)]\,, \tag{5.26}$$

$$x = 1 - [y^2(1-p_+)^2 + (1-y)^2 p_+^2 + 2y(1-y)p_+(1-p_+)]\,, \tag{5.27}$$

$$\frac{1}{4} = (x + p_+ - 2xp_+)(y + p_+ - 2yp_+)\,, \tag{5.28}$$

are obtained. This system can be solved numerically, and the critical value is found to be $p_+ = p_{\mathrm{cn}} = 0.0657$. For p_+ below this value a perturbation at one node will die out and all nodes will again oscillate with period 2. Above this value, attractors must be longer than 2.

5.6 Numerical test of the newly found transitions

Numerics for the case $p_+ = 0$

The numerical experiments presented in Tab. 5.3 show that in the limit of large network sizes there is indeed only one attractor for f_7-networks $\langle A \rangle = \mathrm{med}(A) = 2$ for $N \to \infty$. For small systems, fixed points can appear, see $\min(A) = 1$ for $N = 25$. This is due the fact that a node is allowed to have two inputs coming from the same node.

F is a quantity which is easy to obtain, while for the real number of frozen nodes a full state space search is necessary. The proportion F of nodes which are frozen on a single randomly chosen attractor converges to zero limit of large networks, see Tab. 5.3. The occurrence of some frozen nodes ($\max(F) > 0$) is a finite-size effect, as the standard deviation, $\sigma(F)$, decreases monotonously for $N > 75$.

Confirmation of the transitions at p_{cb} and p_{cn}

In order to identify the transition at p_{cn}, the median attractor length was measured. Sampling the entire state space is impossible already for networks much smaller than the N considered here. This means that attractors are weighted with the size of their basin of attraction. The time evolution is limited to 5000 computational steps for both the transient and the attractor length, in order to keep the overall simulation time within reasonable limits (order of days). This makes it impossible to calculate mean attractor lengths, therefore the median was evaluated.

As shown in Fig. 5.6, with increasing network size the transition becomes increasingly sharp. Below p_{cn}, the fraction of attractors of length 2 converges to some non-zero value with increasing system size, indicating that cycles of length 2 are stable. Above the transition, the median attractor length increases more and more rapidly with increasing system size, indicating a diverging median. Another finding is that attractors become again shorter as the critical point $p_+ = 1/2$ is approached.

The transition at p_{cb} is a transition from an oscillating to a stationary behaviour of b_t. This is obtained from the analytical calculation above. In order to see this transition in the computer simulation, the frequency of phase jumps in b_t on the attractors has been evaluated. An oscillation with period 2 is identified by observing that the value of b_t alternated between being larger and smaller than its mean value $\langle b \rangle$. Deviations from this regular oscillation are "phase jumps", where b_t is two times in a row above or below the mean value. A finite proportion of phase jumps means that the regular oscillation with period 2 ceases to exist and that the point p_{cb} has been reached. The result is shown in the lower panel of Fig. 5.6. With increasing system size, there is an increasingly sharp transition at p_{cb} between zero phase jumps and a finite proportion of phase jumps. In order to verify that b becomes stationary above the transition, its standard deviation Δb was also measured. Δb decreases with increasing system size above p_{cb} while it increases below the transition. For instance, for networks with only 100 nodes it is $\Delta b < 0.1$ for $p_+ > 0.15$. This finding agrees with the analytical result that b is constant in time above p_{cb}.

Table 5.3: Simulation results for f_7-networks. Statistical quantities for the attractor length A and the number of frozen nodes F on one attractor are listed. Beside the extremal values $\min(\cdot)$ and $max(\cdot)$, the median values $\mathrm{med}(\cdot)$, the average $\langle\cdot\rangle$ and the standard deviation $\sigma(\cdot)$ are evaluated for 500 different realizations with 5 random initial conditions for various system sizes.

N	med(A)	min(A)	max(A)	$\langle A \rangle$	$\sigma(A)$	med(F)	min(F)	max(F)	$\langle F \rangle$	$\sigma(F)$
25	2	1	14	2.146	0.821	0	0	25	1.255	4.087
50	2	1	12	2.043	0.485	0	0	50	0.573	3.475
75	2	1	10	2.007	0.184	0	0	75	0.582	3.711
100	2	2	6	2.003	0.098	0	0	41	0.329	1.932
125	2	2	4	2.001	0.040	0	0	56	0.204	1.786
150	2	2	8	2.004	0.133	0	0	65	0.247	1.986
200	2	2	2	2.000	0.000	0	0	26	0.126	1.032

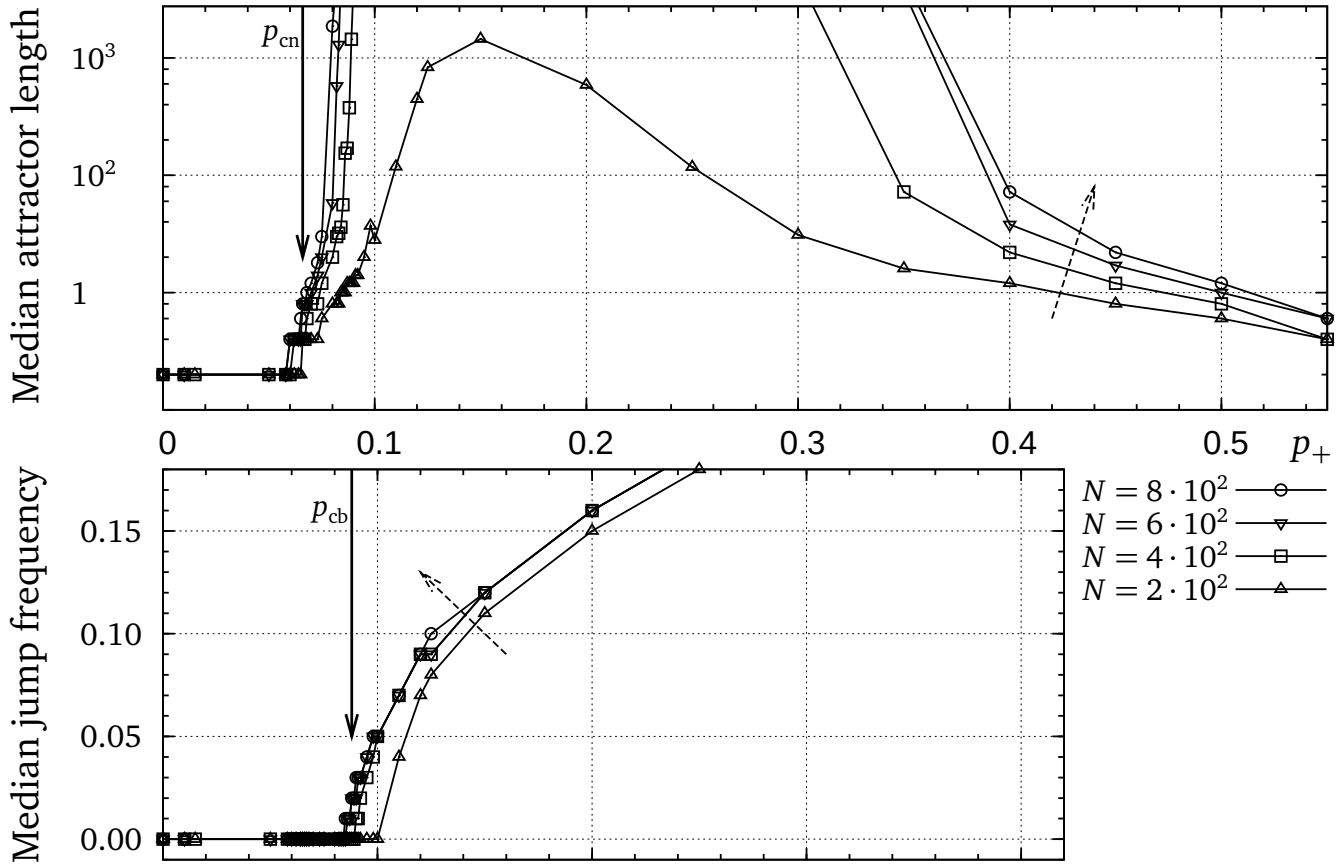


Figure 5.6: Numerical verification for the transitions at p_{cn} and p_{cb}. Both transitions are marked by vertical arrows. The upper panel shows the median attractor length, the lower panel the median frequency of phase jumps on each attractor candidate in dependence on p_+. Each data point corresponds to 5000 sample networks of size N with fixed p_+ and two initial conditions per realization, the dashed arrows show the trend for growing system sizes.

5.7 Summary

Threshold functions are used not only in the context of genetic networks [84, 107, 32] but also in neural networks and in models for markets [118]. When measurements do not make a statement about the actual coupling functions of a real (genetic) network, usually threshold functions are assumed, compare Sec. B. In the present chapter, it was shown that the simplest threshold random Boolean Network shows three different types of phase transitions and not just the generally expected transition between a frozen and a chaotic phase. For parameter values $p_+ < p_{cn}$, all nodes oscillate stably with period two. For $p_{cn} < p_+ < p_{cb}$, the fraction of "on"-nodes oscillate with period two, but attractors are longer. For $p_{cb} < p_+ < 1/2$, the dynamical behaviour is chaotic in the classical sense. For $p_+ > 1/2$, the network is in the frozen phase. The different types of dynamical behaviour are summarised in Fig. 5.7.

The lesson to be learned from this chapter is that the dynamical behaviour of random Boolean networks can be much richer than expected from simple considerations. Some of the networks show global oscillations with higher periods or period doubling cascades in the temporal behaviour of b_t, as was found in special networks where all nodes with the same number of inputs are assigned the same function. Even more interestingly, within a regime with a fixed oscillation

period of the number of "on" nodes, further phase transitions can be hidden. One transition occurs between attractors of period 2 and phase with very long attractors. For the ensemble of biased functions, $\mathcal{F}_p$, no oscillations occur. More generally, the proportion of "on"-nodes can oscillate if the outputs ± 1 in the truth table are not treated equally.

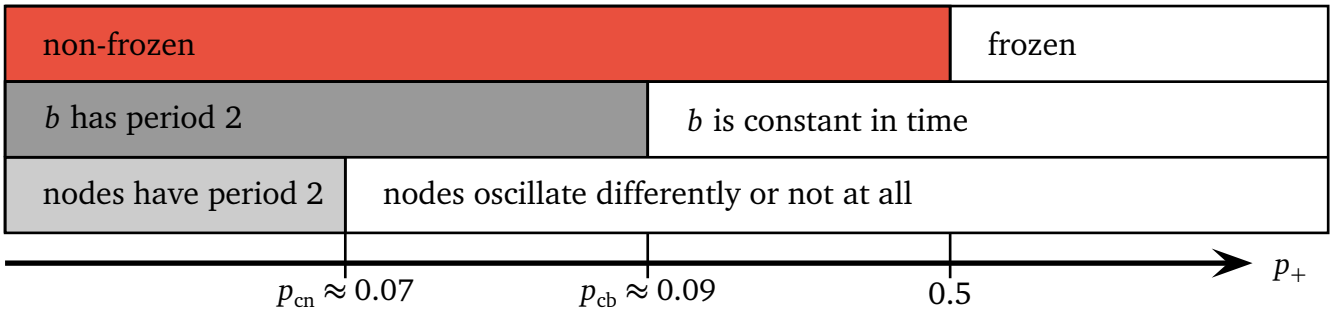


Figure 5.7: Phase diagram for threshold Boolean networks. Even the simplest version with only one parameter p_+ exhibits a rich dynamic behaviour of the model. There are three different types of phase transitions.

Real genetic networks might therefore also have a richer dynamical behaviour than the dynamical classes identified by STUART KAUFFMAN [67, 69]. If the simple classification into "frozen", "critical" and "chaotic" networks fails already in the random threshold model, it will be even less suitable for real genetic networks, which have attractors with very specific properties related to the function of the network.

6 Varying the topology

All Boolean networks considered up to now are variations of the N-k-model which assign a fixed number of inputs to each node. This in-degree $k \leq 2$ has been small in the networks considered in all previous chapters. The number of outputs is Poisson distributed as this distribution results when incoming links are connected at random.

In the present chapter, the in-degree can be larger and the in-degree k_i of a node i is not constant any more (this can be obtained by choosing an according function f_i). A maximal degree, $k_{\max}$, is still given and the term N-$k_{\max}$-model makes sense. A huge amount of networks occurring in nature are known to have a broad degree distribution, which is often well described by a *power law*, see Sec. 6.1.

After shortly reviewing what is known about scale-free networks and their dynamics, the variation to be considered in this chapter is defined. A recently developed stochastic process [72] will be modified and used to determine the relevant nodes of the network by determining stepwise the frozen nodes for the new model variation. Again, the goal is to make a statement about the properties of the system's attractors. The theoretical predictions about the distribution of the nodes at the end of the process will be verified numerically.

6.1 Introduction to scale free networks

In numerous real-world networks one observes that the degree distribution, namely the number of nodes with exactly k links, follows a power law, $P(k) \sim k^{-\beta}$ with $\beta > 1$. An overview over empirically studied scale-free networks and their most important parameters is provided in Tab. 6.1. In particular, the dependence of the cutoff on the system size, $k_{\max}(N)$, is an important parameter for scale-free networks with finite number of nodes N. There are some counter examples of real networks with other fat-tailed degree distributions, such as the grid of power lines in North America [4], and the world-wide air transportation network [13].

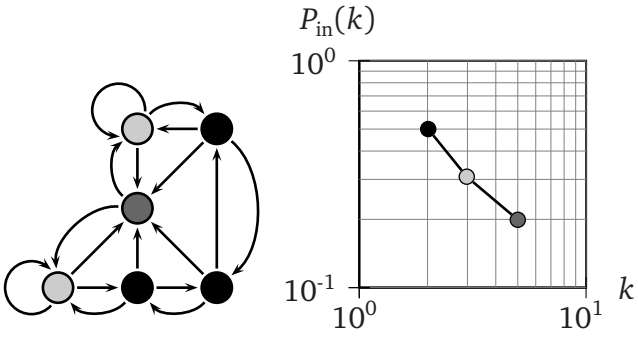


Figure 6.1: Cartoon of a scale-free network (left). The degree-distribution $P_{\text{in}}(k)$ for that network is shown on the right. A power law would give a straight line in a double-logarithmic plot over multiple decades which only makes sense for large system sizes N.

As already mentioned, a power law degree distribution suggests to abandon the idea of a characteristic number of inputs, this is why the term *scale free* was coined. In contrast to that, the bell-shaped degree distribution of random networks has a maximum at the average node degree, $\langle k \rangle$, the *characteristic scale* of the degree. In networks of nodes with scale-free degree distributions, low degrees occur most often, but there are also some very highly connected nodes, the *hubs*. It is necessary that the degree

Table 6.1: Collection of empirical examples for scale-free networks. The data is compiled from a long enumeration in [2], see references within. As most networks are undirected a single exponent β for the degree distribution is sufficient, $P(k) \sim N^{-\beta}$. A detailed description of the systems can be found in Sec. 2.1.

Network	network size N	mean degree $\langle k \rangle$	maximal degree $k_{\max}$	β
WWW	$10^6 \ldots 10^8$	$4.5 \ldots 7.5$	$\approx 10^3$	$1.9 \ldots 2.7$
Internet (hardware)	$10^3 \ldots 10^6$	$2.6 \ldots 3.8$	$30 \ldots 60$	$2.1 \ldots 2.5$
Movie actors	10^6	28.8	900	2.3
Co-authorship	$10^5 \ldots 10^6$	$4 \ldots 173$	$120 \ldots 1100$	$1.2 \ldots 2.5$
Foodwebs	10^2	$4.8 \ldots 8.7$	≈ 30	≈ 1.1
Words in English	$10^5 \ldots 10^6$	$14 \ldots 70$	not given	$2.7 \ldots 2.8$

distribution spans several orders of magnitude, otherwise there a characteristic scale persists. In this sense, the in-degree distribution in Fig. 6.1 is a caricature.

In the present chapter critical networks are considered which have an in-degree distribution $P_{\text{in}}(k)$ that follows a power law,

$$P_{\text{in}}(k_{\text{in}}) \quad \propto \quad k_{\text{in}}^{-\gamma} \qquad \text{with } k_{\text{in}} \in [2, k_{\max}(N)] \,. \tag{6.1}$$

The exponent for the in-degree distribution of a directed network is termed γ and is to be distinguished from the exponent β of an undirected network.

Only the case $\gamma > 2$, where the distribution can be normalised, is studied. In the case $2 < \gamma < 3$, the second moment of the degree distribution diverges, and it has been argued in [83] that this changes the dynamical properties. The out-degree distribution is Poissonian with the mean

$$\langle k_{\text{out}} \rangle \quad = \quad \sum_{k_{\text{out}}} k_{\text{out}} \cdot P(k_{\text{out}}) \,. \tag{6.2}$$

For the considerations below only the in-degree, k_{in}, and its distribution $P_{\text{in}}(k_{\text{in}})$ are required, thus the index "in" will be omitted in the following.

Generating scale-free networks is already a challenge, this issue will be treated next. However, the huge field of network constructing methods should just be touched here as the focus lays on the dynamics on networks.

Growing artificial scale-free networks

Albert-László Barabási and Réka Albert became famous for finding that (undirected) scale-free networks can be generated by a *growth* process with *preferential attachment*, cf. review [6]. The original variant of their algorithm, the so-called *BA-model*, produces networks always having the same exponent, $\beta = 3$. Nowadays there is a zoo of extended versions which allow different exponents or even other network features to be adjusted (such as correlations, see below). The genuine procedure to obtain a BA-network is the following:[1]

1. The growing process starts with a with a small number m_0 of initially unconnected nodes. At every time step a new node $\star$ is attached to the network, such that there are $m \leq m_0$ edges linking $\star$ to already existing nodes.

[1] Note that t is the time of the construction process **of** the network and has nothing to do with the dynamics **on** the network.

2. Each node i already present in the system is chosen with a probability

$$a_i(k_i) \quad = \frac{k_i}{\sum_{i=1}^{m} k_i} \tag{6.3}$$

to attach to node $\star$. Nodes with higher degree k_i are preferred.

The procedure leads to a network with $N = t + m_0$ nodes and $(m \cdot t)$ edges at time t. A drawback of the preferential attachment mechanism is that the new node has to have full information on the degrees of the existing nodes. Furthermore, the BA-method only produces a specific subset of scale-free networks exhibiting inherent correlations [125]: The nodes generated earlier during the process on the one hand tend have a high degree and on the other hand will be more interconnected among themselves than with "younger" nodes. Correlations between the degree of neighbouring nodes can affect the dynamics on these networks [124].

Other ways of generating scale-free networks

One fundamentally different approach to generate a scale-free network is to start directly with the desired number of nodes N and to start connecting them in an intelligent manner such that one eventually ends up with a given degree distribution. One of those algorithms is the *configuration model* (CM) which resembles the scheme introduced by Michael Molloy and Bruce Reed [96]. It is a two step algorithm starting from a given degree distribution $P(k)$ and a set of N nodes. The version for undirected networks consists three steps:

1. To each of the N nodes, a degree k is assigned according to $P(k)$. This number can be understood as the number of *half-links* a node has. Two half-links will later on form a real edge in the final graph. For now, each node has "arms" which have to be connected in the next step.

2. Two free arms from two distinct nodes $\bullet$ and $\circ$ are chosen at random. By connecting them, an edge is established if and only if there is not yet a link between $\bullet$ and $\circ$.

3. Step 2 is repeated as long as it is possible to be performed.

Both self-edges and multiple edges are implicitly forbidden by this algorithm. Furthermore, it is probable that a graph with multiple components arises. If the maximal degree $k_{\max}$ is only limited by the system size N, *hubs* with an extremely high degree lead to correlations – a node with a high degree will most of the times be connected to nodes with low degree. Such networks are called *dissasortative*.

Mathematically, the *assortativity* of a network is measured by the correlation coefficient r between pairs of nodes. When $r = 1$ for a given network, it is said to be perfectly *assortative*, for $r = -1$ the network is completely *disassortative*. In the limit of large networks and with the joint degree distribution $P(j,k)$, that a node of degree j is connected with a node of degree k, holds

$$r = \frac{\sum_{j,k} j \cdot k \cdot \big(P(j,k) - P_e(j)P_e(k)\big)}{\langle k^2 \rangle - \langle k \rangle^2} \qquad \text{with } P_e(k) = \sum_j P(j,k)\,. \tag{6.4}$$

In this definition, $P_e(k)$ is the distribution over edge ends. For undirected networks, the joint degree distribution is symmetric, $P(j,k) = P(k,j)$.

By introducing a cutoff $k_{\max} \propto \sqrt{N}$ for the CM-model as done by MICHELE CATANZARO and coworkers [36] correlations can be avoided, $r \approx 0$. Such networks can then be used as a null model when comparing different real-world networks.

For the approach used in the present chapter the network is not be explicitly constructed but the degree distribution itself is used as initial condition, compare Sec. 6.2.

6.2 Boolean dynamics on a scale-free network

Especially biological networks are known to have a broad degree distribution, which is often well described by a power law (see [2] and references therein). For this reason, several recent studies were devoted to Boolean dynamics on scale-free networks. The majority of these studies used a scale-free in-degree distribution and a Poissonian out-degree distribution. Computer simulations showed that attractors are shorter and frozen nodes occur more often in critical scale-free networks as compared to RBNs with a fixed number of inputs, given the same total number of links and of nodes [51, 75].

KAZUMOTO IGUCHI and coworkers [64] studied a quenched Boolean model on a directed scale-free network by numerical means. Because of the usual constrains of brute force methods they dealt with small networks and a tiny set of initial conditions for each network size (5000 samples only). They mainly offer exponential fittings for the median, mean and the standard deviation of the attractor lengths. For networks with $N \approx 150$ all three quantities grow exponentially with N for $\langle k \rangle = 2$. For $N \sim 10^3$ nodes, the median asymptotically changes from algebraic to exponential type for when approaching $\langle k \rangle = 2$.

Another observation as published in [8, 75] is that attractors are sensitive to perturbations of highly connected nodes, but not of sparsely connected nodes. These and other [114, 111] simulation results are merely stated.

The data has not yet been embedded into an analytical framework. Analytical results obtained so far are limited to calculating the phase diagram of the dynamics using the annealed approximation [8, 9, 52]; only the work by DEOK-SUN LEE and HEIKO RIEGER [83] goes further. They calculate the asymptotic Hamming distance in the chaotic phase and by extrapolating the results to the critical point using a finite-size scaling ansatz in combination with the calculation of the size distribution of perturbed clusters.

The aim of the present section is to offer an analytical approach to random Boolean networks with scale-free input distributions at the critical point. Scaling laws for the number of non-frozen and relevant nodes are obtained. They explain the findings of computer experiments as stated in literature. Furthermore, the results convey a clear understanding of the properties of attractors in these systems.

Two types of defining the in-degree distribution $P(k)$

Two ways of creating the input distribution to a given γ will be explored. The probability, $P(k)$, that a node has exactly k inputs is

$$P(k) = A \cdot k^{-\gamma} \quad \text{with} \quad A = \frac{1}{\sum_{k=k_{\min}}^{k_{\max}} k^{-\gamma}}, \tag{6.5}$$

the normalisation constant A depends on the minimal and the maximal in-degree. A natural choice is $k_{\min} = 1$ which guarantees that there are no unconnected nodes. In the limit of large

systems, $N \to \infty$, the normalisation constant is related to the Riemann Zeta function, $\zeta(\gamma) = \sum_{k=1}^{\infty} k^{-\gamma}$, if $k_{\max} = \infty$, the distribution can only be normalised for $\gamma > 1$,

$$A = \frac{1}{\zeta(\gamma)} \quad \text{for} \quad k_{\max} \to \infty \,. \tag{6.6}$$

Depending on how $k_{\max}$ is chosen, there are two ways to define the in-degree distribution:

1. The maximal occurring in-degree is set to $k_{\max} = N$, the number of nodes in the system. The total number of links and the largest value of k_i differ in this case between different network realizations.

2. The number of nodes with k inputs can be set to be $N \cdot P(k)$ (rounded to the nearest integer). This gives a distribution $P(k)$ that has a cutoff at $k_{\max} \sim N^{1/\gamma}$.

In part of the above-mentioned studies about the dynamics on scale-free networks, the networks were generated using a constraint that does not allow multiple connections between the same nodes, or using a preferential-attachment algorithm.

Furthermore, let each node i have l_i outgoing links. Then, the out-degree distribution $Q(l)$ gives the probability that a node has exactly l outputs. There have to be as many out-links as in-links,

$$\sum_{i=1}^{N} l_i \;\; = \;\; \sum_{i=1}^{N} k_i \,. \tag{6.7}$$

This trivial relation proves useful for the implementation of the stochastic process described in Sec. 6.3.

Choosing the Boolean functions

Different ways of assigning the Boolean functions to the nodes in scale-free networks are checked in the present work. The first way is to study the ensemble of biased functions, $\mathscr{F}_p$. The parameter p is again the *bias*, assigning to each of the 2^{k_i} input configurations the output 1 with a probability p and the output 0 with a probability $1-p$. The value of p was chosen such that the network is critical, i.e., that $p = 1/\langle k \rangle$ [8]. The main results did not depend on whether the exact mean (which can be different for each network), or the theoretical mean $\sum_k kP(k)/N$ was chosen.

The second way of assigning the Boolean functions is to take only constant and reversible functions generating an ensemble $\mathscr{F}_{\mathrm{rc}}$. For a node with any number of inputs, there are two *constant functions*, which fix the value of the node to either -1 or 1, irrespective of its input value. Similarly, for each value of k, there are two *reversible functions*, which are defined by the condition that changing the value of one input always changes the output. A node with a reversible function becomes frozen if and only if all of its inputs are frozen. Such a network is critical if the total number of nodes equals the total number of inputs to nodes with reversible functions. Links to nodes with constant functions have no effect and can be omitted, so that the total number of links becomes identical to the total number of nodes in this case.

6.3 The stochastic process to determine the frozen core

Tamara Mihaljev and Barbara Drossel presented in [92] scaling properties for a general class of critical random Boolean networks with arbitrary but fixed in-degree, $k_i \equiv k$ for all nodes i.

In the thermodynamic limit of large system sizes N, the scaling relations for the number of non-frozen and relevant nodes have been calculated analytically by the authors. The method is based on a stochastic process (as proposed in [72]) that gradually builds up the *frozen core* of the network, i.e. those nodes which are frozen on all attractors.

In the present work, the method from Ref. [72] is adjusted in order to determine the size of the frozen core of critical networks with scale-free input distributions. The frozen core is determined starting from the nodes which are already frozen and by stepwise determining all those nodes that become frozen as a consequence of their inputs being frozen. The main idea of the method is to not specify the network in advance, but to choose the connections within the network while determining the frozen core.

For pedagogical reasons, the following analytical calculations are presented in a form appropriate for a simpler way of choosing the Boolean functions, $\mathscr{F}_{\mathrm{rc}}$, compare Sec. 6.2. The generalisation to other cases is straightforward. At the end of the calculations, it will be outlined how they can be modified to apply to the ubiquitous first ensemble $\mathscr{F}_p$.

It is evident from the way in which the results follow from the calculations (and as critical behaviour is universal in general) that the results are valid for other ensembles, $\mathscr{F}$. A boundary condition for choosing the ensemble $\mathscr{F}$ is that the set used for a node with k inputs becomes identical to the set used for a node with $(k-1)$ inputs when a randomly chosen input is frozen. Of course, the parameters of the ensemble have to be chosen such that the network is critical.

Containers with a certain number of inputs

According to the number j of (non-frozen) inputs, each node can be sorted into a *container* C_j. In each container C_j there are $|C_j(t)|$ nodes with the same number of inputs j at time t of the stochastic process (which is different from the time used for the dynamics **on** the network eventually leading to attractors). In container C_0 the frozen nodes are accumulated. Three different reasons can cause a node to be frozen:

1. A node is *statically frozen by its function* if the output of the function is the same for all possible input combinations. The probability for a node with k inputs to be frozen by its function scales as $2^{-2^{k-1}}$ as there are two homogeneous (and therefore frozen) functions out of the 2^{2^k} possible Boolean functions.

2. A node may be blinking at the beginning but later become *dynamically frozen by its inputs* if all variables of the Boolean function remain constant. This is the case if a node has no non-frozen inputs and would also be put into container C_0.

3. Finally, a node can also be *dynamically frozen* by being part of a so-called *self-freezing loop* as described by Kaufman [72]. This case is not as intuitive as the two previously described ones: Each node's value in a loop (optionally with additional inputs from outside the loop) stays constant in time if it canalizes the value of the succeeding node once the value of the latter settles on its majority bit. The majority bit is the output in the truth-table that occurs most often. This case is not significant for the present system.

The maximal number of non-frozen inputs, $k_{\max}$, determines the number of containers, $j \leq k_{\max}$. Therefore, the largest container index is $k_{\max} = N$ or $k_{\max} \sim N^{1/\gamma}$, depending on the method chosen for creating the input distribution.

Initially, all N nodes of a given network realization are distributed into containers C_j according to the distribution function, $P(j)$, of the in-degrees. This initial container filling can be either

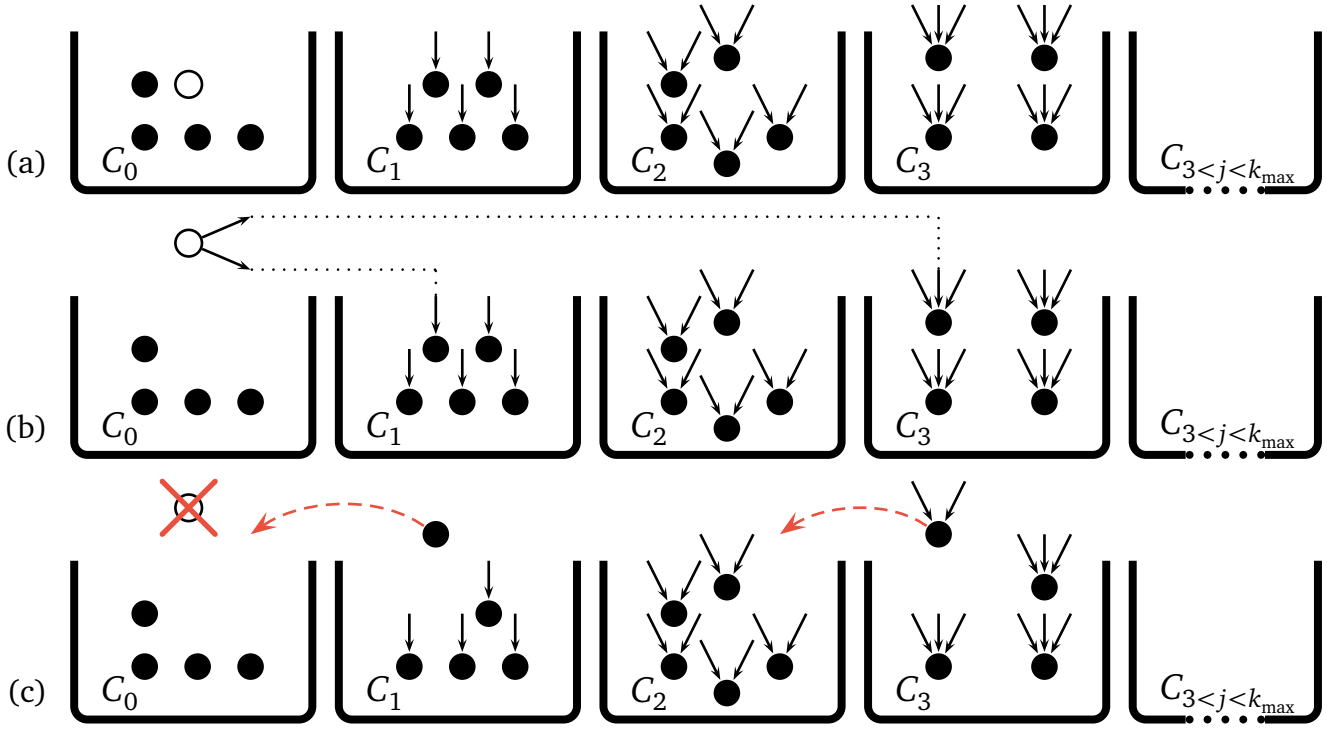


Figure 6.2: Sketch of the stochastic process to determine the frozen core. For a given filling of the containers, a node ∘ from C_0 is chosen (a). The node has $l_\circ$ outputs according to the out-degree distribution $Q(l)$. From the half-links in container C_1 to $C_{k_{\max}}$, the same number $l_\circ$ of inputs are chosen (b). These are connected to the outputs of ∘, sorted into containers according to their number of remaining inputs, and node ∘ is removed from the system (c).

performed by an own stochastic process or by just calculating $P(j)$ and taking the integer value, these are the two methods described in Sec. 6.3.

Moving nodes between the containers

The stochastic process gradually moves nodes from one container to another, most are also removed from the system. The process (as sketched in Fig. 6.2) consists of the following procedure for each time step:

1. A new step of the process starts with taking out a single frozen node ∘ from container C_0,

$$|C_0(t+1)| \;=\; |C_0(t)| - 1\,, \tag{6.8}$$

   which is going to be connected to other nodes during this step of the process, cf. Fig. 6.2(a). Note again, that t is the time of the process.

2. Next, the set $M(t)$ of nodes for which the node ∘ serves as an input are determined. Obviously, the probability for a node ⋆ to be connected to ∘ is higher the more inputs node ⋆ has. The cardinality of the input-set, $|M(t)|$, of the node ∘ obeys the out-degree distribution $Q(l)$.

3. The effect of node ∘ on the nodes $M(t)$ differ. This effect is explored for each node $\star \in M(t)$ separately. (The dotted lines in Fig. 6.2(b) highlights the nodes to which node ∘ serves as input.)

- A node with only one input, $\star \in C_1(t)$, will directly freeze because connecting $\circ$ to $\star$ removes the last non-frozen input. Node $\star$ is then moved to container C_0.
- In general a node $\star$ may originate from a "higher" container, $\star \in C_j(t)$ with $j > 1$. Then, node $\star$ freezes with a *freezing probability* s_j (to be evaluated next), otherwise it just looses one (non-frozen) input and is thus put into container C_{j-1}.

As it is possible to define $s_0 \equiv 1$, this part of a process step can be formulated as:

$$|C_j(t+1)| = |C_j(t)| - 1 \text{ for } j > 1 \text{ and} \tag{6.9}$$

$$|C_x(t+1)| = |C_x(t)| + 1 \text{ where } x = \begin{cases} 0 & \text{with probability } s_j \\ j-1 & \text{with probability } (1-s_j)\,. \end{cases} \tag{6.10}$$

Nodes which are directly moved to the container C_0 during the process are *dynamically frozen*, the others just move to the container with the next smaller index, see dashed lines in Fig. 6.2(c).

4. Finally, a time step of the process is completed by removing node $\circ$ from the system and by adjusting the number of nodes in the process,

$$N(t+1) = N(t) - 1 \qquad \text{with} \qquad N(t=0) \equiv N_{\text{ini}}\,. \tag{6.11}$$

5. The iteration is repeated as long it is possible to create links, the exit condition is

$$\sum\nolimits_{j>0} |C_j(t+1)| = 0 \qquad \text{or} \qquad |C_0(t+1)| = 0\,. \tag{6.12}$$

Note, that up to the end of the present chapter, N_{ini} will denote the system size (in number of nodes) of a given network realization. This convention is to avoid confusion, as $N(t)$ denotes the network size at a given time t of the process.

After executing the process either *all* nodes belong *frozen core* of the network (if $\sum_{j>0} C_j = 0$) or the set of all relevant nodes (if $C_0 = 0$) is left. Both cases are helpful for the goal of finding properties of the attractors.

In the first case, $\sum_{j>0} C_j = 0$, the entire network freezes, and the dynamics of the system runs to the same fixed point for all initial conditions. In the second case, $C_0 = 0$, there is a set of non-frozen nodes. In order to determine the topology of the non-frozen part of the network, one can then fix the connections that have not yet been determined by connecting the remaining inputs at random to the remaining nodes. Once this topology is fixed, it is possible to determine which nodes of the non-frozen ones are relevant. The rest is just to determine the attractors of the dynamics from the relevant components.

The freezing probability s_j for a node in C_j to move directly to C_0

In the process the *freezing probability* s_j appeared accounting for the possibility that an additional frozen input to freeze the whole node. In other words, s_j determines how likely it is that a node never changes its value again if exactly one of its j inputs is frozen. It is possible to derive an analytical expression for s_j for the ensemble $\mathscr{F}_p$ of biased functions.

The freezing probability of a node depends on the number of non-frozen inputs j and on the bias p. For a given exponent γ of the scale-free in-degree distribution the bias is adjusted to its critical value p_c, e.g. [8], determined by

$$2 \cdot p_c \left(1 - p_c\right) \underbrace{\sum_{k=1}^{N} k \cdot p_{\text{in}}(k)}_{\langle k \rangle} . \tag{6.13}$$

Let $p_{\text{nf}}(x)$ be the probability that a node with x inputs has a non-frozen function. As there are two constant Boolean functions, all but two functions are non-frozen. Thus, one needs to subtract the probability for those two functions from 1,

$$p_{\text{nf}}(x) \equiv 1 - \underbrace{\left(p^{2^x} + (1-p)^{2^x}\right)}_{p_{\text{f}}} , \tag{6.14}$$

the expression in the brackets is the probability p_f for a homogeneous outcome in the truth table.

Next, a node $\star$ is considered which has j inputs and where one is connected to the frozen node $\circ$. By definition, those Boolean functions freeze node $\star$ and always lead to the same output for all different input combinations of the $(j-1)$ non-frozen inputs, given that node $\circ$ is fixed to either to 0 or 1. In other words, the case that the overall function (with j inputs) of node $\star$ is frozen is not an event contributing to s_j. All combinations with one frozen input-node have to be excluded as this case would be counted twice otherwise, the probability for these combinations is $p_{\text{nf}}(j-1)$.

The basis of comparison is the probability for overall non-frozen nodes, $p_{\text{nf}}(j)$. Thus, the following expression for s_j is obtained, the second equal sign is due to Eq. (6.14),

$$s_j(p) \;=\; \frac{\pi(j) - \pi(j-1)}{\pi(j)} = \frac{p^{2^{j-1}} + (1-p)^{2^{j-1}} - p^{2^j} - (1-p)^{2^j}}{1 - p^{2^j} - (1-p)^{2^j}} . \tag{6.15}$$

6.4 Analytical prediction for the final container content

In the simpler version of the stochastic process (with $\mathscr{F}_{\text{rc}}$) all freezing probabilities vanish, $s_j \equiv 0$ for all container indices j. It follows immediately that a non-frozen node can only move to the subsequent container with one in-coming half-link less than before during the process. Networks with $\mathscr{F}_{\text{rc}}$ can be adjusted to criticality by setting the number of frozen nodes in the beginning equal to the overall amount of links.

The present section will now focus on an analytical calculation to predict the mean number of nodes remaining in the different containers at the end of the stochastic process.

6.4.1 Mean number of nodes in container C_k

The goal is to evaluate the mean number of nodes in container C_k at the moment where only the fraction $\varepsilon = N/N_{\text{ini}}$ nodes are left in the system. The nodes which are in container C_k at that moment either have been there since the beginning or they have been moved into C_k from a

higher container. In other words, container C_k contains all nodes that initially had $u \geq k$ inputs, and where $(u-k)$ inputs have already become frozen.

In this context, the probability that an input has not yet become frozen during the process is a helpful parameter. That probability is identical to ε, the fraction of nodes which are left in the system. The proof is easy: Only the proportion ε of nodes have not yet been removed from the system, and since an input is connected to every node with the same probability, ε is also the probability that an input has not yet become frozen.

Since container C_u contained initially $\propto N_{\mathrm{ini}} u^{-\gamma}$ nodes (according to the scale-free in-degree distribution), the mean number of nodes in container C_k can be written for the time N of the process as

$$|C_k(N)| \;\propto\; \sum_{u=k}^{k_{\max}} |C_u(0)| \cdot \left(\frac{N}{N_{\mathrm{ini}}}\right)^k \left(1-\left(\frac{N}{N_{\mathrm{ini}}}\right)\right)^{u-k} \binom{u}{k} = N_{\mathrm{ini}} \sum_{u=k}^{k_{\max}} u^{-\gamma} \varepsilon^k (1-\varepsilon)^{u-k} \binom{u}{k} \quad (6.16)$$

For small ε, i.e. near the end of the stochastic process, the nodes currently being in container C_k originated from containers C_u with $u \gg k$. Now, Eq. (6.16) can be approximated by setting $u-k \approx u$ and by replacing the sum with an integral. Furthermore, for small x holds $e^{-x} \simeq (1-x)$ and the binomial coefficient is estimated as u^k. Altogether this leads to

$$|C_k| \;\propto\; N_{\mathrm{ini}} \varepsilon^k \int_k^{k_{\max}} u^{k-\gamma} e^{-u\varepsilon} \mathrm{d}u \quad (6.17)$$

as an approximate expression for the mean number of nodes in container C_k. When evaluating this integral, different cases have to be considered. In the case that the integral is dependent on the cut-off $k_{\max}$, it is left to decide which factor in the integral of Eq. (6.17) determines its value and $|C_k|$ scales as

$$|C_k| \sim \begin{cases} N_{\mathrm{ini}} \cdot \varepsilon^k & \text{for } k < \gamma - 1 \\ N_{\mathrm{ini}} \cdot \varepsilon^{\gamma-1} & \text{for } k > \gamma - 1 \text{ and } \varepsilon^{-1} < k_{\max} \\ N_{\mathrm{ini}} \cdot \varepsilon^k \cdot k_{\max}^{k-\gamma+1} & \text{for } k > \gamma - 1 \text{ and } \varepsilon^{-1} > k_{\max}\,. \end{cases} \quad (6.18)$$

In the first case, the integral is independent of the cutoff. I the second case, the exponential function determines the cutoff to the integral, while in the last case the $k_{\max}$ does.

The variance of the number of frozen nodes

The stochastic process ends when no nodes are left in container C_0. Since the network is critical by definition, on an average, the number of nodes in container C_0 is identical to the number of non-frozen inputs minus the number of non-frozen nodes,

$$\left|C_0(t)\right| \;=\; \sum_{j=1}^{k_{\max}} \left|C_j(t)\right| \cdot j - \sum_{j=1}^{k_{\max}} \left|C_j(t)\right| \quad (6.19)$$

If the stochastic fluctuations during the process are neglected, the number of nodes in container C_0 becomes zero at the same time when the number of nodes in container C_k with $k > 1$ becomes zero, i.e. when $\varepsilon = 0$. However, stochastic fluctuations will terminate the process at earlier

times, at the moment where the fluctuations of the number of frozen nodes become of the same order as the expected number of frozen nodes.

Now, the variance of the number of frozen nodes is evaluated. The probability ε that a given input has not yet become frozen at the moment where N nodes are left in the system, is ε is helpful for that. When ε is small, the number of non-frozen inputs is Poisson distributed, with the variance being identical to the mean, and thus

$$\langle N_{\mathrm{nf}} \rangle = \mathrm{Var}(N_{\mathrm{nf}}) \sim N_{\mathrm{ini}} \cdot \varepsilon \,. \tag{6.20}$$

For small probabilities ε, the vast majority of non-frozen inputs is found in container C_1. A node $\star$ in container C_1 will move to C_0 if $\star$'s remaining input also becomes frozen during the process. It follows that the variance of the number of frozen nodes is also of the order $N_{\mathrm{ini}} \cdot \varepsilon$. The typical fluctuations in the number of frozen nodes are therefore of the order $\sqrt{N_{\mathrm{ini}} \cdot \varepsilon} = \sqrt{N}$.

At the end of the stochastic process, N is identical to the number of non-frozen nodes, N_{nf}. Equating the fluctuations for the number of frozen nodes with the expected number of nodes in C_0 which in turn is of the same order as the expected number of nodes in C_2, one obtains

$$|C_2| \sim \sqrt{N_{\mathrm{nf}}} = \sqrt{N_{\mathrm{ini}} \varepsilon} \,. \tag{6.21}$$

6.4.2 Dependence on the parameters k_{max} and γ

Depending on the value of the in-degree exponent γ and on the choice of k_{max} (either $\propto N$ or $\propto N^{1/\gamma}$), the number of non-frozen nodes scales in a different way with N_{ini}.

Scale-free exponent $\gamma > 3$

For $\gamma > 3$, the first of the three cases in Eq. (6.18) applies to $|C_2|$. Solving Eq. (6.21) for N_{nf} leads to the scaling of the non-frozen nodes at the end of the stochastic process,

$$N_{\mathrm{nf}} \quad \sim \quad N_{\mathrm{ini}}^{2/3} \qquad \text{for } \gamma > 3 \,. \tag{6.22}$$

This is the same result as for random Boolean networks with fixed (effective) in-degree k. Whenever the input distribution $P(k)$ has a finite second moment, $\sum_k P(k) \cdot k < \infty$, the number of non-frozen nodes scales as $N_{\mathrm{ini}}^{2/3}$. The number of non-frozen nodes with two non-frozen inputs scales as

$$|C_2| \quad \sim \quad N_{\mathrm{ini}}^{1/3} \,. \tag{6.23}$$

The number of non-frozen nodes with more than two (non-frozen) inputs, $|C_{>2}|$, depends on whether $k < \gamma - 1$, but it is in any case much smaller than $|C_2|$ and is not further evaluated here.

Scale-free exponent $2 < \gamma \leq 3$

When $2 < \gamma \leq 3$ the choice of k_{max} becomes important. For $k_{\mathrm{max}} \propto N_{\mathrm{ini}}$, the second case of Eq. (6.18) applies and leads to

$$N_{\mathrm{nf}} \quad \sim \quad N_{\mathrm{ini}}^{(2\gamma-4)/(2\gamma-3)} \qquad \text{for } 2 < \gamma \leq 3 \qquad \text{and } k_{\mathrm{max}} \propto N_{\mathrm{ini}} \,. \tag{6.24}$$

For $\gamma = 3$, the exponent of N_{ini} is 2/3, just as for $\gamma > 3$. For smaller values of γ, the exponent decreases to 0 as γ approaches 2. The other way of choosing the initial container configuration corresponds to a fixed input distribution for each realization. Now, $2 < \gamma \leq 3$ in connection with $k_{\mathrm{max}} \propto N_{\mathrm{ini}}^{1/\gamma}$, the third case in Eq. (6.18), is true which means

$$N_{\mathrm{nf}} \quad \sim \quad N_{\mathrm{ini}}^{2\gamma/(\gamma+6)} \qquad \text{for } 2 < \gamma \leq 3 \text{ and } k_{\mathrm{max}} \propto N_{\mathrm{ini}}^{1/\gamma} \,. \tag{6.25}$$

6.4.3 Generalisation to other ensembles of Boolean functions

The results obtained in Sec. 6.4.1 and Sec. 6.4.2 are also valid for other distributions of Boolean functions, $\mathscr{F}$, which are tuned to the critical point. For instance, when biased Boolean functions are chosen, $\mathscr{F}_p$, the freezing probabilities s_j that a node with $j > k$ inputs becomes frozen when $(j-k)$ inputs are frozen are given by Eq. (6.15), a node with more than one input can become frozen when one of its inputs freezes. From networks with fixed in-degree it is already known that the critical behaviour does not depend on the specific choice of functions as long as the network is critical. More generally, critical behaviour is usually universal and does not depend on microscopic details.

Therefore, the Eqs. (6.21) – (6.25) still hold. For the case $\mathscr{F}_p$ the expression Eq. (6.16) for $|C_k|$ gains an additional factor $1-p_f^k$ where p_f is given by Eq. (6.14) to be $p_f = p^{2^k} - (1-p)^{2^k}$ as explained above. The parameter p is again the truth table bias as given by the ensemble $\mathscr{F}_p$. The factor p_f is never close to 0 and therefore does not change the scaling behaviour of the integral in Eq. (6.17).

6.5 Computer simulations of the stochastic process

The analytical considerations of Sec. 6.4 are tested by computer simulations. They have been carried out, both for an ensemble $\mathscr{F}_p$ with biased functions, and with only constant and reversible functions, $\mathscr{F}_{\mathrm{rc}}$.

The probability distribution $P(|C_x|)$ for the non-frozen nodes ($x = \mathrm{nf}$) at the end of the process and the number of nodes in container 2, ($x = 2$), are plotted in Figs. 6.6 – 6.10. double-logarithmically. In particular, both axis are rescaled by $N^{a(x,\gamma,k_{\max})}$ where $a(x,\gamma,k_{\max})$ corresponds to the exponents given in Eqs. (6.21) – (6.25). The distribution is monotonically decreasing in all cases with a characteristic cutoff due to the finite system size. The quality of the data collapse[2] is equally good in all cases. That is true for both ways of choosing the input distributions, $k_{\max} \propto N^x$ with $x \in \{N^{1/\gamma}, N\}$, see Sec. 6.1. The analytical calculations are therefore confirmed by Figs 6.6 – 6.10.

The stochastic process as described in Sec. 6.3 is easily implemented, however, there are some caveats in the detail which are considered in the present section.

General concept of the implementation

For a given exponent γ of the in-degree distribution $P(k)$ and for a given (initial) system size N_{ini}, various network realizations r are generated, the stochastic process is iterated as long as possible and it averages over the resulting final container contents of a single realization. The flowchart Fig. 6.3 visualises this.

The outcome of the simulation is a histogram of how often a certain filling of a given container occurs. The relative number of occurrences can be used as an approximation for the probability distribution if the sampling is long enough, i.e., if the number of realizations is large. The distributions for the final content of the smallest containers, $P(|C_0|)\ldots P(|C_5|)$, is taken into

[2] Data collapse is a way of extracting scaling exponents as occurring for example in phase transitions.

account for the statistics. Beside that, most importantly the distribution for the number of non-frozen nodes, $P(N_{\mathrm{nf}})$, where

$$N_{\mathrm{nf}} = \sum_{k=1}^{k_{\max}} |C_k| \tag{6.26}$$

is measured. All figures show rescaled histograms of the final distributions $P(|C_x|)$ and similarly for $C_{\mathrm{nf}} \equiv N_{\mathrm{nf}}$. In order to obtain meaningful results, data binning is used. The original data falling into in a given small interval, a *bin*, are replaced by a value representing the whole interval. The bin sizes grow with an factor of 1.2, i.e., the next higher bin has a width 1.2 times larger than the current bin.

The goal is of course to sample as many realizations r such that the histograms become smooth to allow statements on the whole ensemble of networks. The simulations show that $r \sim 10^5$ is sufficient. The simulation takes the longer the smaller the exponent γ is as such networks consist of more nodes with many inputs. The shown histograms are evaluated for different system sizes each and they all have the same shape. After rescaling the curves for the various N with the analytically predicted exponent they all coalesce for a given C_k.

Loop over different network realizations

Initialise the container content, $C_j(t = 0)$ for all j

While there are still frozen nodes, $|C_0| > 0$, and unconnected half-links, $\sum_{j>1} |C_j|$

Perform the stochastic process as sketched in Fig. 6.2

Do statistics for the realization (such as determining $N_{\mathrm{non-frozen}}$)

Figure 6.3: Flowchart of the implementation. The skeleton consists of two nested control structures. In the main step (inner part), the contents of the containers changes with time. Stepwise, all inputs which come from a frozen node are removed. The outer part generates the ensemble of networks.

6.5.1 Generating a Poissonian out-degree distribution $Q(l)$

The natural way to generate a Poissonian out-degree distribution is to choose each half-link with probability $\mu = 1/N$. The plain way to implement that is to draw a random number, $r \in [0, 1]$, and to pick a link if $r < \mu$. This algorithm has a major drawback of being computationally very costly as a random number has to be drawn for each of the links in each single iteration of the stochastic process. Instead, a different approach is used which uses less random numbers. The concept is to directly jump to the next half-input to be picked, instead of checking each input one by one. That idea is not new, it emerges from the field of random walks but needs to be modified here to suit its purpose.

All half links are successively arranged on a ray, compare Fig. 6.4, such that there are first the $|C_1|$ inputs of the first container, then the $2 \cdot |C_2|$ of the second one, and so on, ending with $k_{\max} \cdot |C_{k_{\max}}|$ of the last occurring container. A step size Δ counts how many inputs are skipped in order to reach the half-link to be connected to the outputs of the frozen node under

consideration. In order to find such a Δ from a given equally distributed random number $\xi \in [0,1]$ a random variable. On the one hand, the probability ρ to choose a half-link after a step of Δ is

$$\rho \quad := \quad (1-\mu)^{(\Delta-1)} \cdot \mu^1 \,, \tag{6.27}$$

as none of the last $(\Delta-1)$ inputs have been chosen and the probability for not choosing a given input is $(1-\mu)$. On the other hand, ρ also represents the probability that the random number is in the interval which corresponds to that event, $\xi \in \left[(1-\mu)^{\Delta}, (1-\mu)^{\Delta-1}\right)$, thus

$$(1-\mu)^{\Delta} \leq \xi < (1-\mu)^{\Delta-1} \quad \Longrightarrow \quad \Delta - 1 < \frac{\log(\xi)}{\log(1-\mu)} \quad \Longleftrightarrow \quad \Delta = \left\lceil \frac{\log(\xi)}{\log(1-\mu)} \right\rceil . \tag{6.28}$$

This expression can be used to directly reach the next random half-link. Generating a Poissonian out-degree distribution is now much more efficient than the plain method. One only has to do some bookkeeping to which container a chosen half-link belongs and whether a given half-link belongs to the same node as the previously chosen half-link.

Bookkeeping of the half-links

The main ingredient of the algorithm to create a Poissonian out-degree distribution is to dice Δ, telling how many inputs have to be skipped. What is left to be determined is to which node the input belongs, this is achieved by storing the current position on the ray of inputs in a variable v. A given input v is part of container w if

$$\sum_{j=1}^{w-1} |C_w| \cdot w < \quad v \quad \leq \sum_{j=1}^{w} |C_w| \cdot w \,, \tag{6.29}$$

which works of course only for $1 \leq w \leq k_{\max}$. The lower boundary of that interval counts the links in all previous container while the upper limit is the index of the last input of the current container. For each v one also needs to check whether two half-links serve as an input to the same node, compare v_5 and v_6 in Fig. 6.4. For this, one calculates the offset u of input v to the next node,

$$u \quad = \quad \left(\left(\sum_{j}^{w} |C_j| \cdot j\right) - v\right) \mod w \,. \tag{6.30}$$

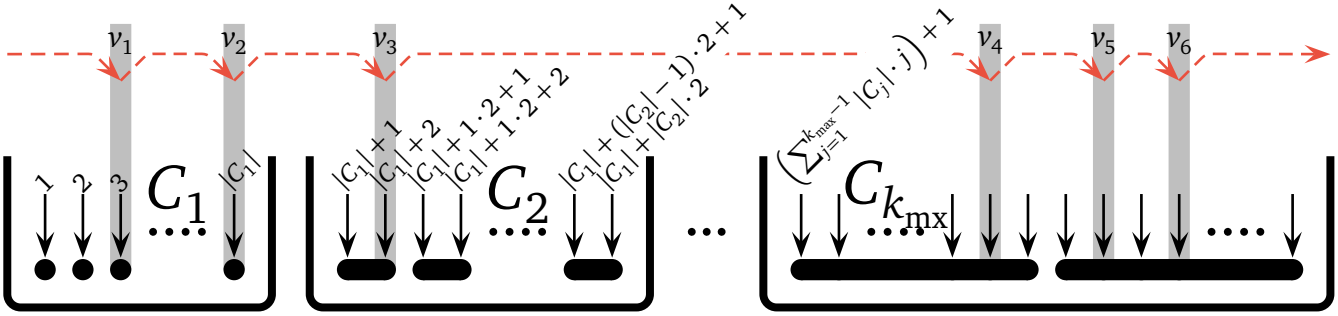


Figure 6.4: Choosing input half-links for a Poissonian distribution. The containers C_1, C_2 up to $C_{k_{\max}}$ are shown, the latter contains two nodes in this example. Some half-links are labelled with their input index. The chosen half-links are marked by gray shaded boxes. The steps Δ are symbolised by dashed arrows.

As long as this skip is smaller than the number of inputs per node in the current container, $\Delta < u$, there is yet another input of the same node picked to be frozen. More generally, that affects the way how a node with w non-frozen inputs is treated in the stochastic process.

6.6 Implications for the dynamics on the network

The results obtained in the previous sections have a variety of implications for the understanding of the dynamics on N-k-networks with a scale-free in-degree distribution. Many properties obtained for critical networks with a fixed number of inputs also apply to scale-free networks, once the frozen nodes have been removed.

Fraction of frozen systems

The fraction of network realizations which will eventually freeze cannot be larger that $1/e$. To show that let there be one node $\star$ with N inputs and all $N-1$ other nodes are already in the frozen container C_0 at the begin of the stochastic process. In this situation the freezing probability is maximal for a given node and link number (which are identical for the ensemble $\mathscr{F}_{\text{rc}}$ adjusted to criticality): The inputs of node $\star$ can only be taken from C_0. The system freezes if none of the half-links takes node $\star$ as input, the probability for that is $\left(1-\frac{1}{N}\right) \approx e^{-1}$ for large N. This is also the upper bound for the fraction of frozen systems in the limit of huge networks. The simulations show that the smaller the exponent of the in-degree distribution is, the larger the system has to be before the limit of large N holds – the dotted line is $1/e$, see Fig. 6.5.

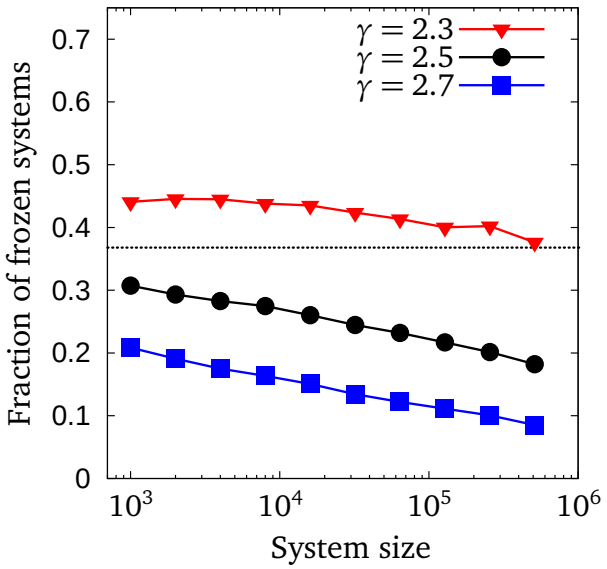


Figure 6.5: Fraction of frozen systems at the end of the process for different γ. The vertical dotted line corresponds to the limit $1/e$.

If a system is completely frozen, the dynamics of a network realisation goes to the same fixed point for all initial conditions if the system is completely frozen. In the limit of large system sizes, $N \to \infty$, the fraction of frozen systems at the end of the process will vanish as by definition the networks are not in the frozen phase but adjusted to criticality, this can also be observed in Fig. 6.5.

Attractor properties

It is known [72] that the number of non-frozen nodes with more than one non-frozen input scales as the square root of the number of non-frozen nodes for critical networks with fixed in-degree,

$$N_{>\,1 \text{ non-frozen inputs}} \sim \sqrt{N_{\text{non-frozen}}}\,. \tag{6.31}$$

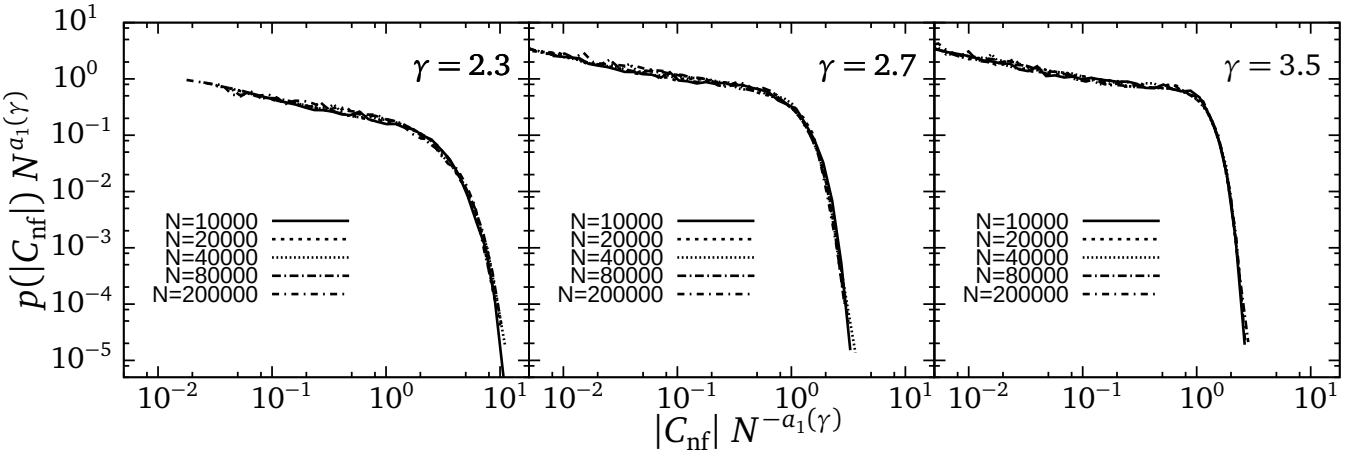


Figure 6.6: Scaling collapse for the number of non-frozen nodes for the ensemble $\mathscr{F}_{rc}$ of frozen and reversible functions. The initial container is drawn at random for every realization, $k_{max} \propto N$. The exponent $a(\gamma)$ is given by Eqs. (6.22) to (6.24), $a(\gamma > 3) = 2/3$ and $a(\gamma \leq 3) = (2\gamma - 4)/(2\gamma - 3)$. For different system sizes (see legend), the histogram includes $5 \cdot 10^5$ realizations. The smaller γ is, the longer the simulations take and the larger the system has to be to show the scaling collapse, compare Fig. 6.5. This can be seen in particular for $\gamma = 2.3$.

Only the dependence of the number of non-frozen nodes on the total number of nodes is changed when $\gamma \in (2,3)$. Therefore, the results obtained in [72] based on these properties of the non-frozen nodes can be taken over.

In particular, it follows that the number of relevant nodes, $N_{relevant}$, in RBNs with a scale-free input distribution (SRBN) scales as

$$N_{relevant} \sim \sqrt{N_{non\text{-}frozen}}\,. \tag{6.32}$$

The number of *relevant components* is of the order of $\log N_{ini}$, with all but a limited number of relevant components being simple loops. This implies again that the mean number $\langle v \rangle$ and length $\langle A \rangle$ of the attractors diverges faster than any power law with the network size,

$$\langle A_{SRBN} \rangle\,, \langle v_{SRBN} \rangle \quad \gtrsim \quad N^x \text{ for any } x\,. \tag{6.33}$$

This explains the finding in [8] that the state-space structure of critical RBNs with fixed k is similar to the states space of a network with a power-law input distribution.

Results for the number of non-frozen nodes

The number of non-frozen nodes decreases with decreasing $\gamma \in (2,3)$, because the exponent in Eq. (6.24) and (6.25) becomes smaller. This explains why several authors have seen more frozen nodes and shorter attractors in scale-free networks as compared to standard RBNs [51, 75].

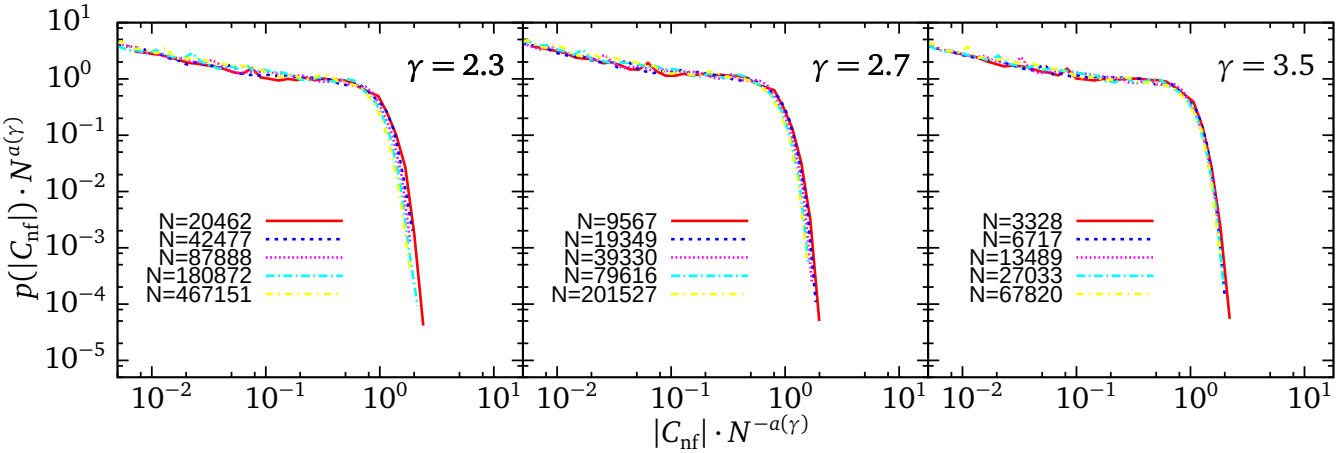


Figure 6.7: Distribution of non-frozen nodes for the ensemble $\mathscr{F}_{\mathrm{rc}}$ of frozen and reversible functions. The initial container filling is the same for all realizations, $k_{\mathrm{max}} \propto N^{1/\gamma}$. As before, $a(\gamma > 3) = 2/3$ but now $a(\gamma \leq 3) = 2\gamma/(\gamma + 6)$, according to Eq. (6.25). Each curve is averaged over $5 \cdot 10^5$ runs of the stochastic process.

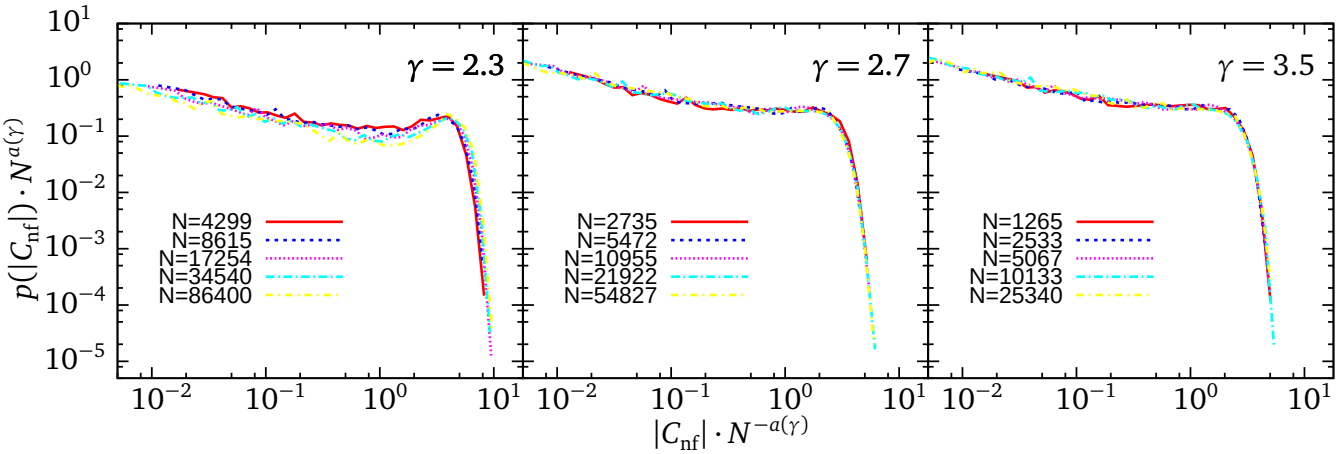


Figure 6.8: Distribution of non-frozen nodes for the ensemble $\mathscr{F}_{\mathrm{p}}$ of functions with bias p. The initial container filling is fixed, $k_{\mathrm{max}} \propto N^{1/\gamma}$, therefore the same expressions as in Fig. 6.7 hold for $a(\gamma)$.

Properties of the attractors

The set of non-frozen and relevant nodes is dominated by nodes with many inputs. This is due to the fact that each input has the same probability of surviving the stochastic process until the end. The average number of inputs of a node that has a surviving link is proportional to $\int k^2 N(k) \mathrm{d}k$, which is dominated by k_{max} for exponents $\gamma \in (2, 3)$ of the in-degree distribution.

When a relevant node is perturbed, the attractor is changed with a large probability. However, when a frozen node is changed, the attractor changes with a probability that vanishes in the limit of large system sizes, $N \to \infty$. This explains the findings in [8, 75] that attractors respond sensitively mainly to perturbations of highly connected nodes.

Discrepancy to previous efforts of an analytical explanation

The analytical results disagree with the finite-size arguments in [83], which predict that the number of non-frozen nodes scales as $N_{\mathrm{ini}}^{(\gamma-1)/\gamma}$ by studying the Hamming distance between two

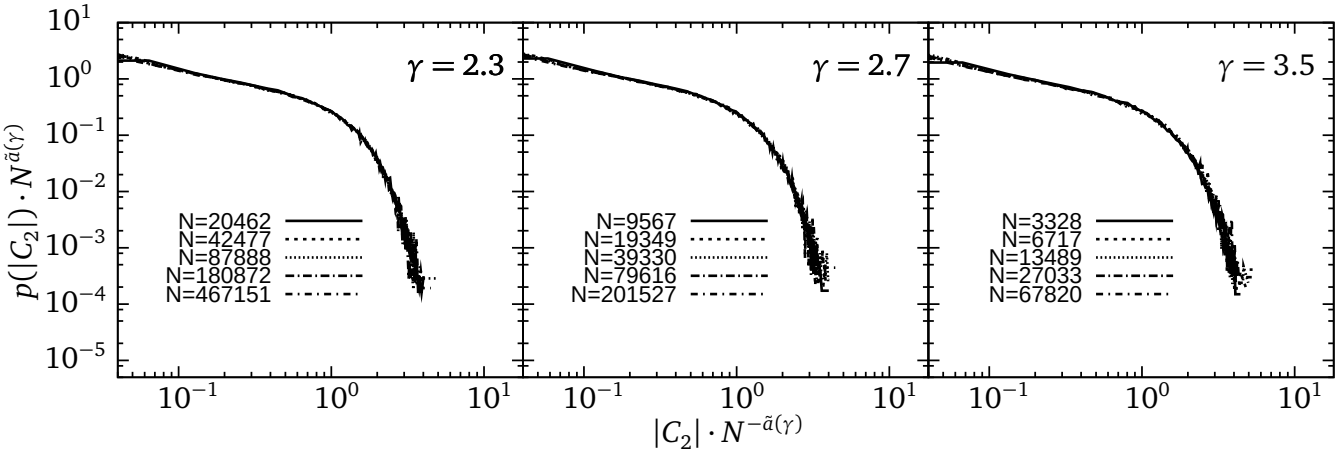


Figure 6.9: Distribution of nodes in container C_2 for the ensemble $\mathscr{F}_{\mathrm{rc}}$ of frozen and reversible functions with fixed initial container filling, $k_{\max} \propto N^{1/\gamma}$. The scaling exponent is given by $\tilde{a}(\gamma) = 1/2 \cdot a(\gamma)$, compare Eq. (6.21).

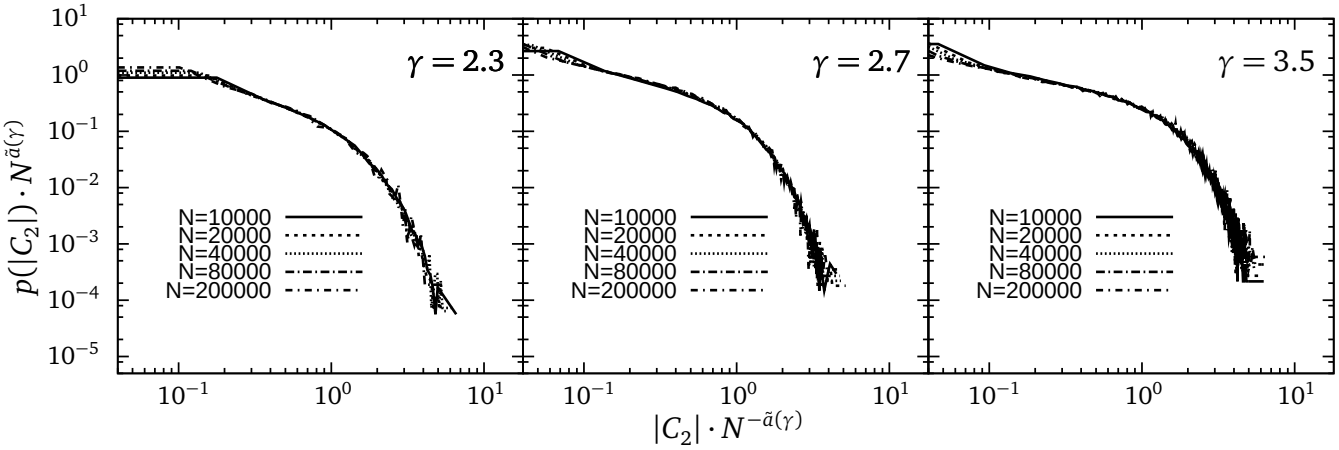


Figure 6.10: Distribution of nodes in container C_2 for the ensemble $\mathscr{F}_{\mathrm{rc}}$ of frozen and reversible functions when the initial container filling is chosen at random, $k_{\max} \propto N$.

evolving network configurations. A possible explanation for the discrepancy is that an infinite (sustained) perturbation as assumed in [83] has properties fundamentally different from those of finite perturbations. In the latter case the arguments based on finite-size scaling do not work.

6.7 Summary

The dynamical properties of critical Boolean networks with power-law in-degree distributions have been studied both analytically and numerically. When the exponent of the in-degree distribution is larger than $\gamma > 3$, the results for the dynamics on the networks are equivalent to those obtained for networks with fixed in-degree: The number of the non-frozen nodes scales with the system size N as $N^{2/3}$. When the exponent of the distribution is between 2 and 3, the number of the non-frozen nodes increases as N^x, with x being between 0 and 2/3 and depending on the exponent and on the cut-off of the in-degree distribution. These and ensuing results explain various findings obtained earlier by computer simulations.

7 Final considerations

A network forms an abstract mathematical model of a complex system with many coupled variables. Such systems include biological systems on various levels of organisation like genetic regulation or metabolic interactions, but also range to social and economic networks. Network science is appealing because of its wide range of applications in different sciences covering problems at vastly different scales. Systems at molecular scale may be modelled by networks as well as social systems or earthquakes. In general, a critical endeavour is to unite the findings in network dynamics of various fields which are separated due to historical reasons into pure mathematics, statistical physics, theoretical computer sciences and quantitative biology. The present work aims to contribute to this.

Boolean networks define the simplest possible dynamics on a network. Due to their simplicity it is possible to investigate extensively the dynamics. Random Boolean networks form the starting point for this thesis. The structure of real networks is, of course, not random, however, random Boolean networks are an important first step on the way to understanding general features of more complex networks and their dynamics. Key characteristics for the dynamics are the number and size of attractors. An attractor is a recurrent set of a network's states. Although the classical model of random Boolean networks with synchronous updating has been introduced in 1969 it took 30 years for an analytical understanding of its key quantities.

The present work starts with considering the attractor distribution of critical networks (Chapter 3) for the classical case. While the mean attractor length and number for this case are mostly understood nowadays, the attractor distribution still lacks analytical understanding which is offered in the present work. The method of relevant components is used to determine the distribution. Then, three new variations of the Boolean network model and their effect on the dynamics are studied — varying the updating scheme (Chapter 4), varying the choice of the functions (Chapter 5) and varying the topology (Chapter 6). The dynamical properties of those variations were not understood before.

Varying the updating scheme. Putting together the knowledge about the statistics of the relevant components and the behaviour of each component allows for statements on the properties of the attractors. For the classical critical random Boolean network (with synchronous updating) it is known that both, the mean number and the length of attractors, diverge faster than any power law with the number of nodes. The dynamics changes for the asynchronous stochastic or a deterministic asynchronous scheme. For the latter case, node based delays are introduced, i.e., some nodes do not always react to its inputs. This increases the attractor lengths and reduces the attractor numbers in comparison to the classical case for deterministic asynchronous updating.

Varying the choice of the functions. The next variation is to restrict the choice of the Boolean update functions and to study the influence on the dynamics. The ensemble under consideration was chosen to be one of Boolean threshold functions. The major result is that the dynamical behaviour is much richer than expected by the usual mean-field considerations: It is shown that within the supposedly chaotic phase oscillations are hidden. An important lesson to learn is

that a more sophisticated way of describing and classifying the dynamical behaviour of Boolean networks is required.

Varying the topology. Last, the focus lays on the effect of the topology on the dynamics. The majority of real-world networks is known to show a power law degree distribution with exponents between 2 and 3. The topology of scale-free networks is a vastly studied subject, however, the (Boolean) dynamics on such networks are not yet understood. The effect of a scale-free in-degree distribution on the dynamics of critical random Boolean networks is investigated. Again, the goal is to analyse the attractors which is performed by using a stochastic process to determine the non-frozen nodes (which are still changing their value on the attractor). If the exponent of the in-degree distribution is larger than 3, the dynamics equals the classical case. When the exponent is between 2 and 3, the number of the non-frozen nodes increases as a power law with the system size with an exponent between 0 and 2/3. This analytically explains earlier numerical findings discussed in literature.

All above considerations where made in the limit of large system sizes. Although Boolean models represent a strong simplification of a much more complex reality, there are several examples where the model correctly captures the essential dynamics of the system. For this reason, the study of random Boolean networks remains an important step on the way towards an understanding of real networks. The present work suggests a strong influence of the implementation details such as the updating scheme, the choice of the functions and the details of the topology on the dynamics. Many real-world network models suffer from the implicit assumption of one of these factors, that is where the importance of the present works lays: One has to be careful what to conclude about the dynamics when studying real-world networks which is therefore part of future projects. Based on these results cooperations with a biomathematical group in Vienna, a molecular genetics group in Mexico and a computing science group in Newcastle is initiated.

A challenge in the field of Boolean networks is now to judge carefully whether the dynamics of real-world network models really reflect properties of the system rather than being artifacts of the network implementation. For this reason it will be helpful to further study other ensembles of random Boolean networks. Those ensembles might be network topologies evolved (for example by an adaptive walk) to be stable against any kind of fluctuations like changing links or update functions. An alternative endeavour is to generate an ensemble of networks from (genetic) databases and to analyse their dynamical properties. A step towards realism is to take the opposite approach of starting with the dynamics and analysing the topology only in the second step.

The discussion of the present work, further corroborated by the ongoing intense collaborations with groups working on widely different networks, proves that much can be gained by the investigation of seemingly simple random Boolean networks.

A Molecular mechanisms of gene regulation

In this chapter the basics of molecular biology will be briefly reviewed. This may be helpful for understanding the original works (e.g. [2, 34] and references within) cited in the present work dealing with empirical biological networks. For this purpose the chemical building blocks are first considered, starting with nucleic acids (Sec. A.1). The level of complexity successively increases when moving via amino acids (Sec. A.2) to proteins (Sec. A.3). The underlying mechanism for constructing proteins within a cell (Sec. A.3.1) is a ubiquitous process. However, it is far from being understood how to predict the structure from a given protein by its genetic blueprint.

A.1 Nucleic acids

In order to explain what nucleic acids are, so-called *sugars* or more formal *saccharides* have to be introduced. They are *carbohydrate* composed by carbon, hydrogen and oxygen. A *monosaccharide* has a chemical formula of the form $(CH_2O)_n$ with some integer n. The most common monosaccharide is *glucose* ($n = 2$), and its isomer[1] *fructose*. In both cases, the C-atoms are arranged in an hexagonal shape. So-called *hydroxyl groups* (-OH) are attached to all carbons except one, the latter is double-bonded to an oxygen forming a *carbonyl group* (=O). Two monosaccharides joined together lead to a *disaccharide*, the most common one is *sucrose* (glucose bonded to fructose). Finally, *polysaccharides* are composed of hundreds to thousands of monosaccharides. Examples are starch, cellulose and glycogen.

The backbone of a *nucleic acid* is made of alternating pentose sugars ($n = 5$) and phosphate molecules ($-PO_4$) joined together in a long chain. To each of the sugars a so-called *nucleotide base* is attached. The main bases are *adenine* a, *guanine* g, *thymine* t, *cytosin* c and *uracil* u, forming the "genetic alphabet". The skeleton of adenine and guanine is a *purine*, the others are based on *pyrimidine*, compare Fig. A.1. Fig. A.1(a) is adeinine for $R_1 = NH_2$ and $R_2 = H$, while it is guanine for $R_1 = H$ and $R_2 = NH_2$. Similarly, Fig. A.1(b) substantiates to thymine for $R_1 = O$ and $R_2 = H$ or to cytosin for $R_1 = H$ and $R_2 = O$. They are all hetereocylic aromatic compounds, i.e., rings of atoms containing at least one carbon and one different element.

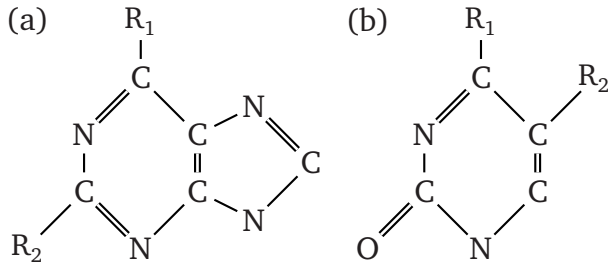


Figure A.1: Purine (a) and pyrimidine (b). The side chains R_1 and R_2 determine the name of the molecule.

Thee are two types of nucleic acids differing in the structure of the sugar. Their roles in gene regulation will be presented in Sec. A.3.1.

[1] An *isomer* is an compounds with the same molecular but different structural formula.

1. The *ribonucleic acid* (RNA) is involved in the synthesis of proteins. Its characteristic sugar is a *ribose*, an aldopentose with $n = 5$ C-atoms and an aldehyde group (-CH=O) at one end. The nucleobases occurring here are $\{c, g, a, u\}$.

2. The *deoxyribonucleic acid* (DNA) contains the programmatic instructions for the cell, one may see it as the recipe how to build the organism. The structural difference to RNA is the presence of a hydroxyl group (-OH) in the sugar, furthermore the base *u* is substituted by the base *t*.

A.2 Amino acids

An *amino acid* has the general formula NH_2-C_αH-R-COOH, where rest R is an organic substituent, a so-called *side-chain*. All amino acids have the hydrogen atom, the *amino group* (-NH_2) and a *carboxyl group* (-COOH) bound to the so-called *alpha carbon* C_α. The detailed chemical properties are listed in Tab. A.1.

Each amino acid is coded in the DNA is by at least one *codon*, a specific sequence of three adjacent nucleotides on a strand of DNA or RNA. Nine of the 20 amino acids used in proteins are essential for humans because the body is not able to synthesise them.[2]

The *polarity* of the amino acid refers to the electrical dipole-moment within the molecule. Polar molecules are solvable in polar solvents like water. Thus, polarity is closely related to *hydropathy* as introduced by Jack Kyte and Russell Doolittle [81]. In this scale, each amino acid has been assigned a value reflecting its relative hydrophilicity and hydrophobicity. If molecules are *hydrophilic*, they are capable of hydrogen bonding, the molecule dissolves in water. The opposite is a *hydrophobic* molecule which often also is *lipophilic* ("fat loving").

Amino acids are joined together through dehydration synthesis to form a *peptide bond* between the carboxyl group (-COOH) of one amino acid and the amino group (-NH_2) of the next one. If the resulting polypeptide chain is short ($\lesssim 30$ amino acids) it is usually called a *peptide*, longer chains are termed *protein*. In biology a polypetide is only called protein if it has a specific function, other fields require the polypetide to have a tertiary structure (to be defined in the Sec. A.3) in order to call it protein.

A.3 Proteins

Proteins are required for the structure, function, and regulation of an organism's cells, tissues, and organs. By weight, proteins account for the major part of the dry weight of cells.

The paradigm formulated by Christian Anfinsen [16], that a protein folds into a *conformation* of lowest free energy without any given help, may not be strictly true. In living cells special proteins called *chaperones* often assist the folding. There are four different levels for a protein's structure:

1. The *primary structure* is a sequence of amino acids. The two ends of the polypetide chain are chemically different, one is carrying a free amino group, NH_3^+, and is the *N-terminus*, the other one is the COO^--end, the *C-terminus*. Alternatively, the N-terminus is often called

[2] The following mnemonics lists all essential amino acids: "Any Help In Learning These Little Molecules Proves Truly Valuable", the one-letter abbreviations from Tab. A.1 are used here.

Table A.1: The 20 canonical amino acids and their properties. The volume can be seen as a scalar proxy quantifying the shape of the side-chain, values taken from [104]. All gray shaded entries in this column are commonly called "small", the lighter shaded one are even said to be "tiny". The number of amino acids with polar side-chains (shaded cells in second last column) is equal with the one with unpolar side-chains. The last column shows the *hydropathy* as given by [81]. Positive values mean that the amino acid is hydrophobic while negative numbers correspond to an hydrophilic side-chain.

Amino acid			Formula	Codon(s)	Side chain properties		
				$*=\{a,g,u,c\}$	vol. [in Å^3]	polarity	Hyd.
Alanine	Ala	A	C_3H_5NO	$gc*$	87.8 ± 2.3	non-polar	1.8
Arginine	Arg	R	$C_6H_{12}N_4O$	$cg*, ag\{a,g\}$	188.2 ± 9.6	polar, positive	-4.5
Asparagine	Asn	N	$C_4H_6N_2O_2$	$aa\{u,c\}$	120.1 ± 4.1	polar, uncharged	-3.5
Aspartic acid	Asp	D	$C_4H_5NO_3$	$ga\{u,c\}$	115.4 ± 2.2	polar, negative	-3.5
Cysteine	Cys	C	C_3H_5NOS	$ug\{u,c\}$	105.4 ± 5.0	non-polar	2.5
Glutamic acid	Glu	E	$C_5H_7NO_3$	$ga\{a,g\}$	140.9 ± 5.3	polar, negative	-3.5
Glutamine	Gln	Q	$C_5H_8N_2O_2$	$ca\{a,g\}$	145.1 ± 5.1	polar, uncharged	-3.5
Glycine	Gly	G	C_2H_3NO	$gg*$	59.9 ± 2.2	non-polar	-0.4
Histidine	His	H	$C_6H_7N_3O$	$ca\{u,c\}$	156.3 ± 6.1	polar, positive	-3.2
Isoleucine	Ile	I	$C_6H_{11}NO$	$au\{u,c,a\}$	166.1 ± 3.4	nonpolar	4.5
Leucine	Leu	L	$C_6H_{11}NO$	$uu\{a,g\}, cu*$	168.0 ± 4.3	non-polar	3.8
Lysine	Lys	K	$C_8H_{12}N_2O$	$aa\{a,g\}$	172.7 ± 5.9	polar, positive	-3.9
Methionine	Met	M	C_5H_9NOS	aug	165.2 ± 1.8	non-polar	1.9
Phenylalanine	Phe	F	C_9H_9NO	$uu\{u,c\}$	189.7 ± 7.4	non-polar	2.8
Proline	Pro	P	C_5H_7NO	$cc*$	123.3 ± 1.8	non-polar	-1.6
Serine	Ser	S	$C_3H_5NO_2$	$uc*, ag\{u,c\}$	91.7 ± 1.8	polar, uncharged	-0.8
Threonine	Thr	T	$C_4H_7NO_2$	$ac*$	118.3 ± 2.3	polar, uncharged	-0.7
Tryptophan	Trp	W	$C_{11}H_{10}N_2O$	ugg	227.9 ± 3.8	non-polar	-0.9
Tyrosine	Tyr	Y	$C_9H_9NO_2$	$ua\{u,c\}$	191.2 ± 8.0	polar	-1.3
Valine	Val	V	C_5H_9NO	$gu*$	138.8 ± 3.6	non-polar	4.2

5' end and the C-terminus just *3' end*. The convention is to present sequences always in N-to-C direction (5'-3' direction).

2. The *secondary structure* consists of regularly repeating local structures stabilised by hydrogen bonds. The two most abundant conformations as presented in Fig. A.2 are:

   a) The *α-helix* resembles a spring. The nitrogen of every building block is hydrogen bonded to the carboxyl group (-C=O) of the amino acid four peptide bonds away. Each amino acid therefore corresponds to a 100° turn and to a translation of 1.5 Å along the helical axis.

   b) The *β-sheet* can form either from neighbouring polypetide chains that run in the same orientations or from a chain folding back and forth upon itself. One of those *β*-strands

is typically 5–10 amino acids long, the peptide backbones are almost fully extended. Again, multiple hydrogen bonds hold the amino acids together.

3. The *tertiary structure* is the overall shape of a single protein molecule. The tertiary structure is generally stabilised by non-local interactions.

4. The *quaternary structure* is the shape resulting from the interaction of several protein molecules, which are subunits of a larger assembly or of a protein complex.

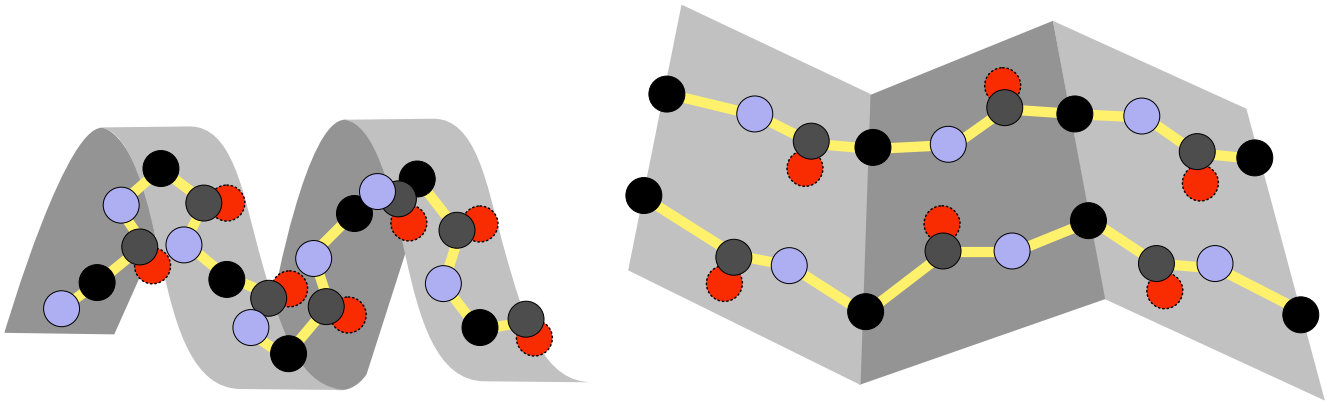


Figure A.2: α-helix (left) and antiparallely pleated β-sheet (right). For the sake of clarity, only the backbone of the polypetide chain (yellow polygon) is sketched. The α-atoms (black dots) would have side chains attached, the other carbon atoms double-bonded to an oxygen atom (gray dots overlapping red dots with dashed border). Finally, there are also nitrogen atoms (blue dots) in the chain.

The primary structure of a protein can readily be deduced from its nucleotide sequence, see Sec. A.3.1. Based on the primary structure, many features of the secondary structure can be predicted with the aid of computer programs. However, predicting protein tertiary structure remains a very tough problem, although some progress has been made in this important area.

A.3.1 Protein biosynthesis

Proteins are encoded in the DNA and built up in all living organisms by a multi-step process, which consists of *transcription* and *translation*. The details are essential for the understanding of empiric gene regulatory networks. The occurring molecules might interfere with another. For the corresponding network abstraction of gene regulation such an interaction would be represented by an edge (depending on the desired level of coarse-graining).

Transcription

During *transcription* the genetic information stored in the double-stranded DNA is transcribed into the short-lived *messenger RNA* (mRNA) that is complementary to one strand of the DNA. An enzyme called *RNA polymerase* binds to a *promoter region*, a special region on the DNA facilitating the transcription of a particular region of DNA, the *gene*. Promoters are usually assumed to be located *upstream* of the gene. Once attached to the DNA, the polymerase moves stepwise along the DNA, unwinding the DNA helix one amino acid site ahead. The RNA chain is extended by one nucleotide at a time. The nucleotides consumed are the triphosphates ATP, UTP, CTP and GTP. More specifically, two additional steps take place:

1. The DNA is usually tightly packed together in a condensed form. Before the RNA polymerase can attach, the DNA needs to be unzipped. Beside other effects, so-called *histones* act as spools, around which the DNA winds. Even the histones play a role in gene regulation.

2. After the completed mRNA copy of a given DNA sequence is released translation can begin. In eukaryotic cells the mRNA first has to move out of the nucleus into the cytoplasm. In prokaryotic cells translation can already start before transcription has been finished.

Translation

The protein is formed only in the *translation* step. The name makes sense as the process is really a translation from one code to another – from the nucleotide sequence to an amino acid sequence. In the simplest case of bacteria the translation takes place at a *ribosome*, a round cellular organelle composed of specialised *ribosomal RNA* (rRNA). A set of three bases defines a *codon*, see Tab. A.1, the *transfer RNA* (tRNA) delivers the amino acid corresponding to a given codon. The secondary structure of tRNA has a typical cloverleaf shape. The ribosome moves along the mRNA, matching codons and adding amino acids stepwise to the growing polypeptide chain. When the machinery reaches a *stop codon* (a triplet $ua\{a,g\}$ or uga), the ribosome releases both the newly built-up polypeptide and the mRNA. The polypeptide forms into its native shape and starts acting as a functional protein in the cell.

B Real-world genetic networks

In statistical physics one usually focuses on an ensemble of network realisations. Those ensembles then fulfil some given criterion such as a certain degree distribution or a certain subclass of allowed update functions for each node. When studying real-world networks it is usually not possible to study ensembles of networks but rather single networks.

This chapter provides a short overview of the widely used experimental methods and a few major techniques for reconstruction of networks. The idea is to sensitise the reader to what kind of systematic errors may occur when simply analysing a given real-world network without taking into account how the particular network was extracted from the data. A few small genetic networks will also be presented as they are more reliably studied by so-called *knock-out experiments* where an organism is engineered to lack the activity of single genes.

B.1 High-throughput biology

The aim of high-throughput biology is to quantify biology. It is not self-evident how to extract a network from bio-chemical data. The workhorse for data acquisition is *high-throughput screening* (HTS). This term summarises methods which produce huge amounts of experimental (raw) data by means of widely automated processes. Using robotics, liquid handling devices and sensitive detectors probes are tested against a library of reaction partners.

The massive amount of data serves as the basis for network models. The most widely used methods to construct networks from the data are based on Bayesian inference, singular value decomposition or mutual information approaches, see review [34].

Data acquisition

A key lab equipment for that is the *microtiter plate* holding up to 9600 small plastic test tubes termed *wells*. Each well has a volume between 0.01 ml to 5 ml where the reactions take place. The miniaturisation are *microarrays*, with them it is possible to survey the expression of more than 10^4 genes per experiment. Such a *DNA-chip* consists of an array of microscopic spots of DNA snippets, each spot contains picomoles of a specific DNA sequence.

The principal method will be demonstrated by describing a *DNA-microarray* experiment where two populations of cells are compared. One population of (yeast) cells is living in an aerobic environment (oxygen is present) while the other one lives under anaerobic conditions. The experiment is divided into several steps:

1. The cells have grown and adjusted to the given environment such that at the end some genes are activated, others are deactivated. By spinning the test tubes (containing the cells populations) with a centrifuge, the cells aggregate at the bottom and the nutrient solution can be removed. The single-stranded *messenger RNA* (mRNA) of the cells is isolated using a chemical, the extraction buffer.

2. Then, all the mRNA molecules are converted to coloured *complementary DNA* (cDNA). This process is similar to the translation in real cells except that to each nucleic acid a certain

dye molecule is attached. After the copying process from mRNA to cDNA is completed, the mRNA is degraded. Let the aerobic population be marked red while the other one is green.

3. Now, cDNA from both populations is combined in a single tube. Every spot on a DNA-chip is then incubated with the mixed cDNA. Some of the colour-labelled cDNA will bind to the DNA snippets on the spots, the rest is washed off the plate.

4. The last step is the measurement. The microarray is scanned with a red and a green laser to detect how much cDNA of a given colour is bound. From the results a merged image is created. All colour data from one spot on the chip is abstracted into a single colour value, ranging between red and green in the current example. A red pixel means that the gene under consideration on a given spot is expressed only in the aerobic environment, while green means that the gene is expressed under anaerobic conditions only. Yellowish values correspond to expression under both conditions as both red and green markers are present.

The results of microarray measurements are so-called expression patterns, they form the basis for quantitative biology. Once the raw data has been acquired, the next question is how to interpret them.

B.1.1 Methods for network reconstruction

The three main reconstruction techniques used in high throughput biology — the Bayesian approach, the mutual information method and Singular Value Decompostion — are introduced in the present chapter.

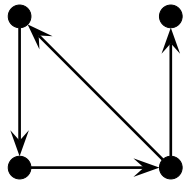


Figure B.1: Hypothetic real-world network to be reconstructed in the following.

To appemphasise the differences between some reconstruction techniques and for pedagogical reasons, the real network topology is assumed to be explicitly known and is given by Fig. B.1. For each method, a figure will show the possible reconstruction results of this network.

Bayesian network approach

A Bayesian network is a graph representing the probabilistic interdependencies of random variables and is based on the theory developed by THOMAS BAYES in 1763. The goal is to find the most probable network given the expression pattern. The directed edges represent direct causal dependencies. By definition only directed acyclic graphs are allowed as resulting networks. This is the major drawback of this approach as feedback structures are essential for gene regulation networks.

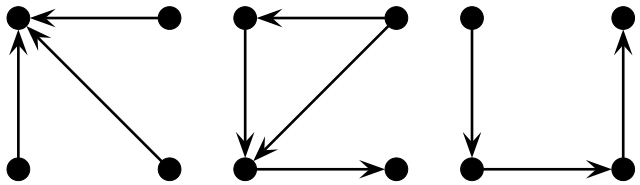


Figure B.2: Bayesian network reconstruction. The last network is the one with highest probability, it lacks the closing of the loop the original network in Fig. B.1 contains.

In order to reconstruct a network from given statistical data, a series of tests is performed to learn more about conditional independence of any pair of genes. If a gene g_1 influences the expression of gene g_3 only indirectly through the influence of g_2 (network: $g_1 \to g_2 \to g_3$), this structure can be recovered if the joint probabilities are known, because for a fixed expression level g_2 the expression level of g_3 is independent of the expression level of g_1.

The basic idea of the network reconstruction algorithm as developed by GEORGE REBANE and JUDEA PEARL [106] is the distinction between the three possible triplets of nodes, g_1, g_2, g_3, in a directed acyclic graph,

$$\begin{array}{lll} g_1 \rightarrow & g_2 & \rightarrow g_3 \\ g_1 \leftarrow & g_2 & \rightarrow g_3 \\ g_1 \rightarrow & g_2 & \leftarrow g_3 \end{array} \qquad \begin{array}{l} p(g_3|g_2)p(g_2|g_1)p(g_1) \\ p(g_3|g_2)p(g_1|g_2)p(g_2) \\ p(g_2|g_1,g_3)p(g_1)p(g_3) \end{array} \tag{B.1}$$

The first two cases represent the same dependencies and are not distinguishable: g_1 and g_3 are independent from each other, given the expression level of g_2. The third case can directly be identified as g_1 and g_3 are (marginally) independent while all other pairs are not.

All three building blocks in Eq. (B.1) have the same *skeleton* (the same undirected graph when leaving away the arrows) but it is possible to identify the direction of the arrows in the last case. The algorithm systematically determines the skeleton of the graph and then orients the edges according to the conditional interdependencies. The results are different network topologies which at the end are judged by how well they explain the data. Fig. B.2 shows three possible Bayesian reconstructions of the network in Fig. B.1.

Networks based on mutual information

The *mutual information* is a measure for the correlation between two random variables. In gene regulatory models, an edge between two genes is established if the mutual information of their expression patterns is larger than some threshold, compare Fig. B.3. The difference to the Bayesian approach is that the mutual information approach decides edge by edge whether or not to include a possible interaction into the network structure. Therefore, this approach is faster than the computationally expensive Bayesian reconstruction method. However, no information about the directionality of the network is recovered using the mutual information method

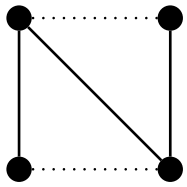


Figure B.3: Mutual information network reconstruction. The edges represent correlations between pairs of variables, but only the solid ones are above the threshold.

Singular value decomposition

In linear algebra *singular value decomposition* (SVD) describes the factorisation of a rectangular matrix. The *singular values* characterise the matrix in the same way as *eigenvalues* do for quadratic matrices. For a matrix S with $m \times n$ entries the decomposition can be written as $A = UDV^T$ (D is a diagonal $m \times n$-matrix containing the singular values, U is quadratic with $m \times m$ and V has $n \times n$ entries).

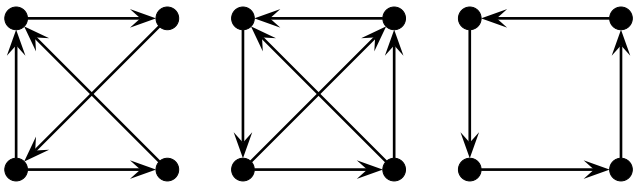


Figure B.4: Networks reconstructed by Singular Value Decomposition (SVD). The sparsest network (on the right) is assumed to be the target network.

In experiments using microarrays, the raw data is preprocessed first in order to reduce the noise. The obtained values serve as input for the SVD which can then detect and extract small

signals. The columns of the left singular vectors U are the so-called *gene coefficient vectors* and the right singular vectors are the rows of V^T, those are commonly known as the *expression level vectors*. The SVD delivers candidate solutions for the network topology. Based on the empirical observation that genetic networks are usually sparse, the solution with the smallest number of edges is chosen to be the most likely network, compare Fig. B.4.

B.2 Networks obtained from detailed genetic experiments

Large networks have a strong statistical uncertainty due to the applied reconstruction methods. Furthermore, reproducibility of the experiments is a huge issue because the effects of small experimental details to the high-throughput screening is not sufficiently explored. Researchers in this field admit that one might see different results for the same experimental setup processed by different robots.[1]

The alternative way to build gene regulation networks from experimental data is to include the detailed molecular-genetics experiments by a comprehensive literature inquiry. The resulting networks usually are rather small but have been thoroughly characterised by laboratories.

Several biological networks exist where a Boolean model faithfully reproduces the known activity sequence of regulatory elements appearing during the cell cycle, a nice starting point are the reviews [7, 50]. Integrating experiments to network building blocks have successfully been carried out for developmental processes of multicellular organisms, e.g. [7, 42, 49, 1, 25] and references within [3, 11]. Some of them will be presented here.

The transcription network of baker's yeast

Yeast is an eukaryotic organism belonging to the kingdom of fungii. *Saccharomyces cerevisiae* is widley known as *baker's yeast* which is used for fabricating bread (as the name suggests) but also for fermenting alcoholic beverages. It is one of the major model organisms in biological research because it is easy to cultivate (e.g., it grows between 10° to 37° C).

In the yeast gene-regulatory network, the individual knockout of each of about 70% of all genes has no effect on the survival of the cell [39]. Fangting Li and coauthors compiled an effective network [84] from a huge collection of publications about the interaction of single genes. The network is much more coarse-grained than the ones compiled from huge databases; out of approximately 800 genes in yeast the authors simplified it to 11 key regulators with Boolean dynamics. Amazingly, it was found that this seemingly crude approximation is sufficient

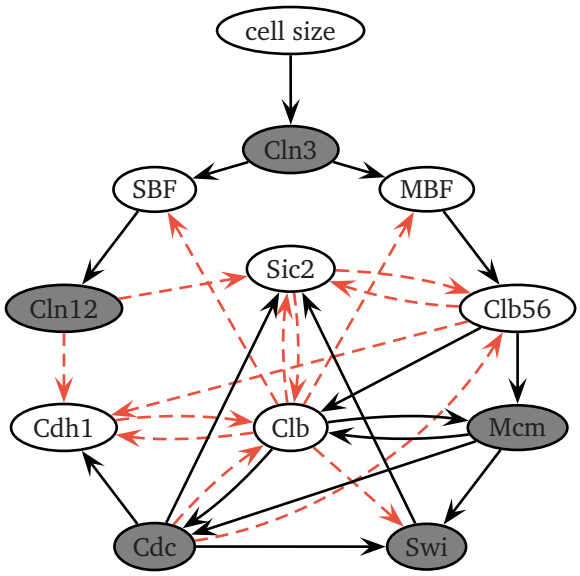


Figure B.5: Simplified version of the baker's yeast transcription network. The interaction is either activating (black solid arrows) or deactivating (red dashed arrows). Some nodes regulate themselves (gray shaded).

[1] Lecture by Orly Alter and personal communication with their coworkers.

to describe the correct succession of events occurring during the cell cycle. For yeast there is strong evidence [43] that this biological trajectory in state space is robust.

Arabidopsis thaliana

The attractors of the networks may correspond to particular cell-types and traits such as the basin's sizes provide valuable information about the robustness of the biological system [35]. The plant *Arabidopsis thaliana*, see Fig. B.6, is another famous model system in biology. *Arabidopsis* is a small flowering plant related to cabbage and mustard and is mostly found at dry and sunny locations. It was the first plant to have a totally sequenced genome[2], see Fig. B.6.

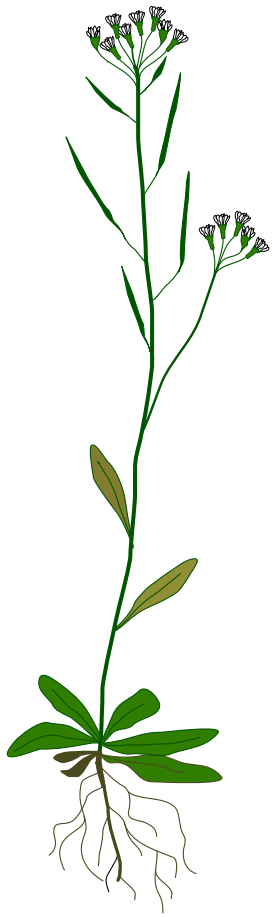


Figure B.6: Arabidopsis thaliana

The first genetic regulatory network studied for *Arabidopsis* is the cell-type determination in developing flowers. A discrete model has been built on the base of continuously emerging experimental data [90, 49, 34, 37]. Interestingly, this model recovers ten attractors of the dynamics which correspond to the cell-types observed in early flower development. These attractors match the gene expression profiles of the so-called inflorescence cells (four attractors), sepal (one attractor), petal (two attractors), stamen (two attractors) and carpel (one attractor) primordial cells. Studying the robustness of this system in the Boolean translation under various updating schemes is work in progress in cooperation with Mariana Benítez.

The second network model for *Arabidopsis* considers cell-types in the leafs and in the epidermis. Again, the model has been built based on the experimental findings [26, 25]. Beside that, root epidermal cell-type determination and spatial patterning of the cell-types have also been in focus [90, 26, 25].

[2] Extensive information about that plant can be found at `http://www.arabidopsis.org`

Bibliography

[1] Abouheif, Ehab and Wray, Gregory A. *Evolution of the Gene Network Underlying Wing Polyphenism in Ants*. Science (July 2002) **297**, pages 249–252. doi:10.1126/science.1071468

[2] Albert, Réka. *Scale-free networks in cell biology*. Journal of Cell Science (November 2005) **118**, pages 4947–4957. doi:10.1242/jcs.02714

[3] Albert, Réka. *Network Inference, Analysis, and Modeling in Systems Biology*. Plant Cell (November 2007) **19**, pages 3327–3338. doi:10.1105/tpc.107.054700. ↗ http://www.plantcell.org

[4] Albert, Réka; Albert, István and Nakarado, Gary L. *Structural vulnerability of the North American power grid*. Phys. Rev. E (February 2004) **69** (2), page 025 103. doi:10.1103/PhysRevE.69.025103

[5] Albert, Réka and Barabási, Albert-László. *Dynamics of Complex Systems: Scaling Laws for the Period of Boolean Networks*. Phys. Rev. Lett. (June 2000) **84** (24), page 5660. doi:10.1103/PhysRevLett.84.5660

[6] Albert, Réka and Barabási, Albert-László. *Statistical mechanics of complex networks*. Rev. Mod. Phys. (January 2002) **74**, pages 47–97. cond-mat/0106096

[7] Albert, Réka and Othmer, Hans G. *The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster*. J. Theo. Bio. (July 2003) **223**, pages 1–18. doi:10.1016/S0022-5193(03)00035-3

[8] Aldana, Maximino. *Boolean dynamics of networks with scale-free topology*. Physica D (October 2003) **185**, pages 45–66. doi:10.1016/S0167-2789(03)00174-X

[9] Aldana, Maximino and Cluzel, Philippe. *A natural class of robust networks*. Proc. Natl. Acad. Sci. USA (July 2008) **100**, pages 8710–8714. doi:10.1073/pnas.1536783100

[10] Aldana-Gonzalez, Maximo; Coppersmith, Susan and Kadanoff, Leo P. *Boolean Dynamics with Random Couplings*. Perspectives and Problems in Nonlinear Science (May 2003) pages 23–89. nlin/0204062, ↗ http://www.fis.unam.mx/~max/

[11] Almaas, Eivind. *Biological impacts and context of network theory*. The Journal of Experimental Biology (May 2007) **210**, pages 1548–1558. doi:10.1242/jeb.003731

[12] Alon, Uri. *An Introduction to Systems Biology: Design Principles of Biological Circuits* (Chapman & Hall, 2006), 1st edition. ISBN 1584886420

[13] Amaral, Luís Antonío Nunes; Scala, Antonio; Barthélémy, Marc and Stanley, Harry Eugene. *Classes of small-world networks*. Proc. Natl. Acad. Sci. USA (October 2000) **97** (21), pages 11 149–11 152. doi:10.1073/pnas.200327197

[14] Andrecut, Mircea. *Mean field dynamics of random Boolean networks*. Journal of Statistical Mechanics: Theory and Experiment (2005) **2005** (02), page P02 003. doi:10.1088/1742-5468/2005/02/P02003

[15] Andrecut, Mircea and Ali, M. K. *Chaos in a simple Boolean network*. Internat. J. of Modern Physics B (2001) **15** (1), pages 17–23. doi:10.1142/S021797920100259X

[16] Anfinsen, Crhistian B. *Principles that Govern the Folding of Protein Chains*. Science (July 1973) **181** (4096), pages 223–230. doi:10.1126/science.181.4096.223

[17] Aracena, Julio. *Maximum Number of Fixed Points in Regulatory Boolean Networks*. Bulletin of Mathematical Biology (July 2008) **70** (5), pages 1398–1409. doi:10.1007/s11538-008-9304-7

[18] Barabasi, Albert-László. *Linked: How Everything is Connected to Everything Else and What it Means for Business and Everyday Life* (Plume Books, 2003), Reissue edition. ISBN 0452284392

[19] Bassler, Kevin E.; Lee, Choongseop and Lee, Yong. *Evolution of Developmental Canalization in Networks of Competing Boolean Nodes*. Phys. Rev. Lett. (July 2004) **93**, page 038 101. doi:10.1103/PhysRevLett.93.038101

[20] Bastolla, Ugo and Parisi, Giorgio. *Closing probabilities in the Kauffman model: An annealed computation*. Physica D (1996) **98** (1), pages 1–25. cond-mat/9803224

[21] Bastolla, Ugo and Parisi, Giorgio. *A Numerical Study of the Critical Line of Kauffman Networks*. J. Theo. Bio. (July 1997) **187**, pages 117–133. doi:10.1006/jtbi.1997.0423

[22] Bastolla, Ugo and Parisi, Giorgio. *The modular structure of Kauffman networks*. Physica D (May 1998) **115**, pages 219–233. doi:10.1016/S0167-2789(97)00242-X

[23] Bastolla, Ugo and Parisi, Giorgio. *The Modular Structure of Kauffman Networks*. Physica D (May 1998) **115** (3&4), pages 219–233. cond-mat/9708214

[24] Bastolla, Ugo and Parisi, Giorgio. *Relevant elements, magnetization and dynamical properties in Kauffman networks. A numerical study.* Physica D (May 1998) **115** (3&4), pages 203–218. ↗ http://chimera.roma1.infn.it/GIORGIO/indexhome.htm

[25] Benítez, Mariana; Espinosa-Soto, Carlos; Padilla-Longoria, Pablo and Álvarez Buylla, Elena. *Interlinked nonlinear subnetworks underlie the formation of robust cellular patterns in Arabidopsis epidermis: a dynamic spatial model*. BMC Systems Biology (2008) **2**, page 98. doi:10.1186/1752-0509-2-98

[26] Benítez, Mariana; Espinosa-Soto, Carlos; Padilla-Longoria, Pablo; Díaz, José and Alvarez-Buylla, Elena R. *Equivalent genetic regulatory networks in different contexts recover contrasting spatial cell patterns that resemble those in Arabidopsis root and leaf epidermis: a dynamic model*. The International Journal of Developmental Biology (2007) **51**, pages 139–155. doi:10.1387/ijdb.062183mb

[27] Berdahl, Andrew; Shreim, Amer; Sood, Vishal; Davidsen, Joern and Paczuski, Maya. *Avalanches, branching ratios, and clustering of attractors in Random Boolean*

Networks and in the segment polarity network of \emphDrosophila (May 2008) ↗ http://arxiv.org/abs/0805.0326

[28] Bhattacharjya, Amartya and Liang, Shoudan. *Power-Law Distributions in Some Random Boolean Networks*. Phys. Rev. Lett. (1996) **77**, page 1644. doi:10.1103/PhysRevLett.77.1644

[29] Bilke, Sven and Sjunnesson, Fredrik. *Stability of the Kauffman model*. Phys. Rev. E (December 2001) **65**, page 016 129. doi:10.1103/PhysRevE.65.016129

[30] Bilke, Sven and Sjunnesson, Fredrik. *Stability of the Kauffman Model*. Phys. Rev. E (2002) **65**, page 016 129. cond-mat/0107035

[31] Boole, George. *The Calculus of Logic*. Cambridge and Dublin Mathematical Journal (1848) **3**, pages 183–198. ↗ http://www.maths.tcd.ie/pub/HistMath/People/Boole/CalcLogic/

[32] Bornholdt, Stefan and Sneppen, Kim. *Robustness as an evolutionary principle*. Proceeedings of the Royal Society B (November 2000) **267** (1459), pages 2281–2286. doi:10.1098/rspb.2000.1280

[33] Bray, A. J. *Universal scaling function for domain growth in the Glauber-Ising chain*. J. Phys. A (1990) **23** (2), pages L67–L72. doi:10.1088/0305-4470/23/2/005

[34] Álvarez Buylla, Elena R; Benítez, Mariana; Dávila, Enrique Balleza; Álvaro Chaos; Espinosa-Soto, Carlos and Padilla-Longoria, Pablo. *Gene regulatory network models for plant development*. Current Opinion in Plant Biology (February 2007) **10**, pages 83–91. doi:10.1016/j.pbi.2006.11.008

[35] Álvarez Buylla, Elena R.; Álvaro Chaos; Aldana, Maximino; Benítez, Mariana; Cortes-Poza, Yuriria; Espinosa-Soto, Carlos; Hartasánchez, Diego A.; Lotto, R. Beau; Malkin, David; Santos, Gerardo J. Escalera and Padilla-Longoria, Pablo. *Floral Morphogenesis: Stochastic Explorations of a Gene Network Epigenetic Landscape*. Public Library of Science (PLOS) ONE (2008) **3**, page e3626. doi:10.1371/journal.pone.0003626

[36] Catanzaro, Michele; Boguñá, Marián and Pastor-Satorras, Romualdo. *Generation of uncorrelated random scale-free networks*. Phys. Rev. E (February 2005) **71** (2), page 027 103. doi:10.1103/PhysRevE.71.027103

[37] Álvaro Chaos; Aldana, Max; Espinosa-Soto, Carlos; de León, Berenice; Arroyo, Adriana and Alvarez-Buylla, Elena. *From Genes to Flower Patterns and Evolution: Dynamic Models of Gene Regulatory Networks*. Journal of Plant Growth Regulation (December 2006) **25**, pages 278–289. doi:10.1007/s00344-006-0068-8

[38] Chmiel, Anna M.; Sienkiewicz, Julian; Suchecki, Krzysztof and Holyst, Janusz A. *Networks of companies and branches in Poland*. Physica A (September 2007) **383** (1), pages 134–138. doi:10.1016/j.physa.2007.04.095

[39] Christensen, Claire and Albert, Réka. *Using graph concepts to understand the organization of complex systems*. International Journal of Bifurcation and Chaos (July 2007) **17** (7). ↗ http://arxiv.org/abs/q-bio/0609036

[40] http://www.citebase.org

[41] Colizza, Vittoria; Barrat, Alain; Barthélemy, Marc and Vespignani, Alessandro. *The role of the airline transportation network in the prediction and predictability of global epidemics*. Proc. Natl. Acad. Sci. USA (February 2006) **103** (7), pages 2015–2020. doi:10.1073/pnas.0510525103

[42] von Dassow, George; Meir, Eli; Munro, Edwin M. and Odell, Garrett M. *The segment polarity network is a robust developmental module*. Nature (July 2000) **406**, pages 188–192. doi:10.1038/35018085

[43] Davidich, Maria I and Bornholdt, Stefan. *Boolean network model predicts cell cycle sequence of fission yeast*. Public Library of Science (PLOS) ONE (2008) **3**, page e1672. doi:10.1371/journal.pone.0001672

[44] Derrida, Bernard and Pomeau, Yves. *Random networks of automata: a simple annealed approximation*. Europhys. Lett. (January 1986) **1** (2), pages 45–49

[45] Derrida, Bernard and Stauffer, Dietrich. *Phase transitions in two dimensionalKauffman cellular automata*. Europhys. Lett. (November 1986) **2** (10), pages 739–745. ↗ http://www.lps.ens.fr/phystat/

[46] Drossel, Barbara. *Number of attractors in random Boolean networks*. Phys. Rev. E (July 2005) **72**, pages 016 110–016 115. doi:10.1103/PhysRevE.72.016110

[47] Drossel, Barbara. *Random Boolean Networks*. In Heinz-Georg Schuster (editor) *Reviews of Nonlinear Dynamics and Complexity* (Wiley, 2008), volume 1, pages 69–110. ISBN 978-3-527-40729-3

[48] Drossel, Barbara; Mihaljev, Tamara and Greil, Florian. *Number and Length of Attractors in a Critical Kauffman Model with Connectivity One*. Phys. Rev. Lett. (March 2005) **94**, pages 088 701–088 704. doi:10.1103/PhysRevLett.94.088701

[49] Espinosa-Soto, Carlos; Padilla-Longoria, Pablo and Álvarez Buylla, Elena R. *A Gene Regulatory Network Model for Cell-Fate Determination during Arabidopsis thaliana Flower Development That Is Robust and Recovers Experimental Gene Expression Profiles*. The Plant Cell (November 2004) **16**, page 2923–2939. doi:10.1105/tpc.104.021725

[50] Faure, Adrien; Naldi, Aurelien; Chaouiya, Claudine and Thieffry, Denis. *Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle*. Bioinformatics (July 2006) **22** (14), pages e124–131. doi:10.1093/bioinformatics/btl210

[51] Fox, Jeffrey J. and Hill, Colin C. *From topology to dynamics in biochemical networks*. Chaos (December 2001) **11**, pages 809–815. doi:10.1063/1.1414882

[52] Fronczak, Piotr; Fronczak, Agata and Holyst, Janusz A. *Kauffman Boolean model in undirected scale-free networks*. Phys. Rev. E (March 2008) **77**, pages 036 119–036 125. doi:10.1103/PhysRevE.77.036119

[53] Gershenson, Carlos. *Updating Schemes in Random Boolean Networks: Do They Really Matter?* In J. Pollack; M. Bedau; P. Husbands; T. Ikegami and R. A. Watson (editors) *Artificial Life IX Proceedings of the Ninth International Conference on*

the Simulation and Synthesis of Living Systems (MIT Press, 2004) pages 238–243. ↗ http://uk.arxiv.org/abs/nlin.AO/0402006

[54] Gershenson, Carlos; Broekaert, Jan and Aerts, Diederik. *Contextual Random Boolean Networks*. In W Banzhaf; T. Christaller; P. Dittrich; J. T. Kim and J. Ziegler (editors) *Advances in Artificial Life, 7th European Conference, ECAL 2003 LNAI 2801* (Springer-Verlag, 2003) pages 615–624. ↗ http://uk.arxiv.org/abs/nlin.AO/0303021

[55] Glauber, Roy J. *Time-Dependent Statistics of the Ising Model*. J. Math. Phys. (February 1963) **4** (2), pages 294–307. doi:10.1063/1.1703954

[56] Golder, Scott; Wilkinson, Dennis and Huberman, Bernardo A. *Rhythms of Social Interaction: Messaging within a Massive Online Network*. In *3rd International Conference on Communities and Technologies (CT2007)* (East Lansing, 2007) ↗ http://www.hpl.hp.com/research/idl/papers/facebook

[57] Greil, Florian and Drossel, Barbara. *Dynamics of Critical Kauffman Networks under Asynchronous Stochastic Update*. Phys. Rev. Lett. (July 2005) **95**, pages 048 701–4. doi:10.1103/PhysRevLett.95.048701

[58] Greil, Florian and Drossel, Barbara. *Kauffman networks with threshold functions*. Eur. Phys. J. B (May 2007) **57** (1), pages 109–113. doi:10.1140/epjb/e2007-00161-0

[59] Greil, Florian; Drossel, Barbara and Sattler, Joost. *Critical Kauffman networks under deterministic asynchronous update*. New Journal of Physics (2007) **9** (10), page 373. ISSN 1367-2630. doi:10.1088/1367-2630/9/10/373

[60] Hallinan, Jennifer and Wiles, Janet. *Asynchronous dynamics of an artificial genetic regulatory network*. In *Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife9)* (Boston, 2004) ↗ http://www.staff.ncl.ac.uk/j.s.hallinan/pubs.html

[61] Harvey, Inman and Bossomaier, Terry. *Time out of joint: attractors in asynchronous random boolean networks*. Proceedings of the Fourth European Conference on Artificial Life (ECAL97) (1997) pages 67–75. doi:10.1.1.48.6693. ↗ http://www.informatics.sussex.ac.uk/users/inmanh/

[62] Holme, Petter; Edling, Christofer R. and Liljeros, Fredrik. *Structure and time evolution of an Internet dating community*. Social Networks (May 2004) **26** (2), pages 155–174. doi:10.1016/j.socnet.2004.01.007

[63] Holme, Petter; Huss, Mikael and Jeong, Hawoong. *Subnetwork hierarchies of biochemical pathways*. Bioinformatics (March 2003) **19** (4), pages 532–538. doi:10.1093/bioinformatics/btg033

[64] Iguchi, Kazumoto; ichi Kinoshita, Shu and Yamada, Hiroaki S. *Boolean dynamics of Kauffman models with a scale-free network*. J. Theo. Bio. (July 2007) **247**, pages 138–151. doi:10.1016/j.jtbi.2007.02.010

[65] http://www.imdb.com

[66] Ingerson, T.E. and Buvel, R.L. *Structure in asynchronous cellular automata*. Physica D (1984) **10** (1–2), pages 59–68. doi:10.1016/0167-2789(84)90249-5

[67] Kauffman, Stuart. *Homeostasis and Differentiation in Random Genetic Control Networks*. Nature (October 1969) **224**, pages 177–178. doi:10.1038/224177a0

[68] Kauffman, Stuart; Peterson, Carsten; Samuelsson, Björn and Troein, Carl. *Genetic networks with canalyzing Boolean rules are always stable*. Proc. Natl. Acad. Sci. USA (December 2004) **101** (49), pages 17 102–17 107. doi:10.1073/pnas.0407783101

[69] Kauffman, Stuart A. *Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets*. J. Theo. Bio. (March 1969) **22**, pages 437–467

[70] Kaufman, Viktor and Drossel, Babara. *On the properties of cycles of simple Boolean networks*. Eur. Phys. J. B (January 2005) **43**, pages 115–124. cond-mat/0410546

[71] Kaufman, Viktor and Drossel, Barbara. *Relevant components in critical random Boolean networks*. New Journal of Physics (2006) **8**, page 228. doi:10.1088/1367-2630/8/10/228

[72] Kaufman, Viktor; Mihaljev, Tamara and Drossel, Barbara. *Scaling in critical random Boolean networks*. Phys. Rev. E (2005) **72**, page 046 124

[73] Kermack, W. O. and McKendrick, A. G. *A Contribution to the Mathematical Theory of Epidemics*. Proceeedings of the Royal Society A (August 1927) **115**, pages 700–721. doi:10.1098/rspa.1927.0118

[74] Kesseli, Juha; Ramo, Pauli and Yli-Harja, Olli. *Iterated maps for annealed Boolean networks*. Phys. Rev. E (October 2006) **74** (4), pages 046 104–14. doi:10.1103/PhysRevE.74.046104

[75] Kinoshita, Shu-Ichi; Iguchi, Kazumoto; Yamada, Hiroaki S.; Tokuyama, Michio; Oppenheim, Irwin and Nishiyama, Hideya. *Robustness of Attractor States in Complex Networks*. AIP Conference Proceedings (February 2008) **982**, pages 768–771. doi:10.1063/1.2897899

[76] Klemm, Konstantin and Bornholdt, Stefan. *Robust gene regulation: Deterministic dynamics from asynchronous networks with delay*. Phys. Rev. E (2005) page 055101

[77] Klemm, Konstantin and Bornholdt, Stefan. *Stable and unstable attractors in Boolean networks*. Phys. Rev. E (November 2005) **72**, pages 055 101–4. doi:10.1103/PhysRevE.72.055101

[78] Klemm, Konstantin and Bornholdt, Stefan. *Topology of biological networks and reliability of information processing*. Proc. Natl. Acad. Sci. USA (December 2005) **102** (51), pages 18 414–18 419. doi:10.1073/pnas.0509132102

[79] Krawitz, Peter and Shmulevich, Ilya. *Basin Entropy in Boolean Network Ensembles*. Phys. Rev. Lett. (April 2007) **98**, pages 158 701–158 704. doi:10.1103/PhysRevLett.98.158701

[80] Kumpula, Jussi M.; Onnela, Jukka-Pekka; Saramaki, Jari; Kaski, Kimmo and Kertesz, Janos. *Emergence of Communities in Weighted Networks*. Phys. Rev. Lett. (November 2007) **99** (22), pages 228 701–228 704. doi:10.1103/PhysRevLett.99.228701

[81] Kyte, Jack and Doolittle, Russell F. *A simple method for displaying the hydropathic character of a protein*. Journal of Molecular Biology (May 1982) **157** (1), pages 105–132. doi:10.1016/0022-2836(82)90515-0

[82] Lacasa, Lucas; Luque, Bartolo; Ballesteros, Fernando; Luque, Jordi and Nuño, Juan Carlos. *From time series to complex networks: The visibility graph*. Proc. Natl. Acad. Sci. USA (April 2008) **105** (13), pages 4972–4975. doi:10.1073/pnas.0709247105

[83] Lee, Deok-Sun and Rieger, Heiko. *Broad edge of chaos in strongly heterogeneous Boolean networks*. J. Phys. A (2008) **41**, page 415 001. doi:10.1088/1751-8113/41/41/415001

[84] Li, Fangting; Long, Tao; Lu, Ying; Ouyang, Qi and Tang, Chao. *The yeast cell-cycle network is robustly designed*. Proc. Natl. Acad. Sci. USA (March 2004) **101**, pages 4781–4786. doi:10.1073/pnas.0305937101

[85] Liljeros, Fredrik; Edling, Christofer R.; Amaral, Luis A. Nunes; Stanley, H. Eugene and Aberg, Yvonne. *The web of human sexual contacts*. Nature (June 2001) **411** (6840), pages 907–908. doi:10.1038/35082140

[86] Liu, Min and Bassler, Kevin E. *Emergent criticality from coevolution in random Boolean networks*. Phys. Rev. E (October 2006) **74**, pages 041 910–5. doi:10.1103/PhysRevE.74.041910

[87] Matache, Mihaela T. and Heidel, Jack. *Random Boolean network model exhibiting deterministic chaos*. Phys. Rev. E (May 2004) **69** (5), page 056 214. doi:10.1103/PhysRevE.69.056214

[88] Matache, Mihaela T. and Heidel, Jack. *Asynchronous random Boolean network model based on elementary cellular automata rule 126*. Phys. Rev. E (February 2005) **71** (2), page 026 232. doi:10.1103/PhysRevE.71.026232

[89] McCulloch, Warren and Pitts, Walter. *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biology (December 1943) **5** (4), pages 115–133. doi:10.1007/BF02478259

[90] Mendoza, Luis and Álvarez Buylla, Elena R. *Genetic Regulation of Root Hair Development in Arabidopsis Thaliana: A Network Model*. Journal of Theoretical Biology (June 2000) **204**, pages 311–326. doi:10.1006/jtbi.2000.2014

[91] Mesot, Bertrand and Teuscher, Christof. *Critical Values in Asynchronous Random Boolean Networks*. Advances in Artificial Life (2003) **2801**, pages 367–376. ISSN 978-3-540-20057-4

[92] Mihaljev, Tamara and Drossel, Barbara. *Scaling in a general class of critical random Boolean networks*. Phys. Rev. E (October 2006) **74**, pages 046 101–10. doi:10.1103/PhysRevE.74.046101

[93] Mihaljev, Tamara and Drossel, Barbara. *Scaling in a general class of critical random Boolean networks*. Phys. Rev. E (October 2006) **74**, pages 046 101–10. doi:10.1103/PhysRevE.74.046101

[94] Milo, R.; Shen-Orr, S.; Itzkovitz, S.; Kashtan, N.; Chklovskii, D. and Alon, Uri. *Network Motifs: Simple Building Blocks of Complex Networks*. Science (October 2002) **298** (5594), pages 824–827. doi:10.1126/science.298.5594.824

[95] Milo, Ron; Itzkovitz, Shalev; Kashtan, Nadav; Levitt, Reuven; Shen-Orr, Shai; Ayzenshtat, Inbal; Sheffer, Michal and Alon, Uri. *Superfamilies of Evolved and Designed Networks*. Science (March 2004) **303** (5663), pages 1538–1542. doi:10.1126/science.1089167

[96] Molloy, Michael and Reed, Bruce. *A critical point for random graphs with a given degree sequence*. In *Random Structures & Algorithms* (1995), volume 6 pages 161—179. doi:10.1.1.24.6195

[97] Moreira, André Auto and Amaral, Luís A. Nunes. *Canalizing Kauffman Networks: Nonergodicity and Its Effect on Their Critical Behavior*. Phys. Rev. Lett. (June 2005) **94**, page 218 702. doi:10.1103/PhysRevLett.94.218702

[98] Nekovee, Maziar. *Worm epidemics in wireless ad hoc networks*. New Journal of Physics (2007) **9** (6), page 189. doi:10.1088/1367-2630/9/6/189

[99] Onnela, Jukka-Pekka; Saramaki, Jari; Hyvonen, Jorkki; Szabo, Gabor; de Menezes, M. Argollo; Kaski, Kimmo; Barabasi, Albert-Laszlo and Kertesz, Janos. *Analysis of a large-scale weighted network of one-to-one human communication*. New Journal of Physics (2007) **9** (6), page 179. doi:10.1088/1367-2630/9/6/179

[100] Paczuski, Maya; Bassler, Kevin E. and Corral, Álvaro. *Self-Organized Networks of Competing Boolean Agents*. Phys. Rev. Lett. (April 2000) **84**, page 3185. doi:10.1103/PhysRevLett.84.3185

[101] Paolo, Ezequiel A. Di. *Rhythmic and non-rhythmic attractors in asynchronous random Boolean networks*. Biosystems (March 2001) **59** (3), pages 185–195. doi:10.1016/S0303-2647(01)00102-2

[102] Paul, Ute; Kaufman, Viktor and Drossel, Barbara. *Properties of attractors of canalyzing random Boolean networks*. Phys. Rev. E (February 2006) **73** (2), pages 026 118–9. doi:10.1103/PhysRevE.73.026118

[103] Peixoto, Tiago P. and Prado, Carmen P. C. *Network of epicenters of the Olami-Feder-Christensen model of earthquakes*. Phys. Rev. E (July 2006) **74** (1), pages 016 126–9. doi:10.1103/PhysRevE.74.016126

[104] Perkins, Stephen J. *Protein volumes and hydration effects*. European Journal of Biochemistry (1986) **157** (1), pages 169–180. doi:10.1111/j.1432-1033.1986.tb09653.x

[105] Raeymaekers, Luc. *Dynamics of Boolean Networks Controlled by Biologically Meaningful Functions*. Journal of Theoretical Biology (October 2002) **218** (3), pages 331–341. doi:10.1006/jtbi.2002.3081

[106] Rebane, George and Pearl, Judea. *The recovery of causal poly trees from statistical data*. In *Proceedings Uncertainty in Artifical Intelligence* (Elsevier Science, Seattle, 1987) pages 222–228

[107] Rohlf, Thimo and Bornholdt, Stefan. *Criticality in random threshold networks: annealed approximation and beyond*. Physica A (July 2002) **310**, pages 245–259. doi:10.1016/S0378-4371(02)00798-7

[108] Rohlfshagen, Philipp and Paolo, Ezequiel A. Di. *The circular topology of rhythm in asynchronous random Boolean networks*. Biosystems (February 2004) **73** (2), pages 141–152. doi:10.1016/j.biosystems.2003.11.003

[109] Samuelsson, Björn and Troein, Carl. *Superpolynomial Growth in the Number of Attractors in Kauffman Networks*. Phys. Rev. Lett. (March 2003) **90**, page 098 701. doi:10.1103/PhysRevLett.90.098701

[110] Samuelsson, Bjørn and Troein, Carl. *Superpolynomial growth in the number of attractors in Kauffman networks*. Phys. Rev. Lett. (2003) **90**, page 098 701

[111] Serra, Roberto; Villani, Marco and Agostini, Luca. *On the dynamics of random Boolean networks with scale-free outgoing connections*. Physica A (August 2004) **339**, pages 665–673. doi:10.1016/j.physa.2004.03.026

[112] Shen-Orr, Shai S.; Milo, Ron; Mangan, Shmoolik and Alon, Uri. *Network motifs in the transcriptional regulation network of Escherichia coli*. Nature Genetics (May 2002) **31** (1), pages 64–68. doi:10.1038/ng881

[113] Shmulevich, Ilya and Kauffman, Stuart A. *Activities and Sensitivities in Boolean Network Models*. Phys. Rev. Lett. (July 2004) **93** (4), page 048 701. doi:10.1103/PhysRevLett.93.048701

[114] e Silva, A. Castro; da Silva, J. Kamphorst Leal and Mendes, J. F. F. *Scale-free network with Boolean dynamics as a function of connectivity*. Phys. Rev. E (December 2004) **70**, page 066 140. doi:10.1103/PhysRevE.70.066140

[115] Sloane, Neil J. A. *The On-Line Encyclopedia of Integer Sequences*. www.research.att.com/~njas/sequences/A003418

[116] Socolar, Joshua E. S. and Kauffman, Stuart A. *Scaling in ordered and critical random Boolean networks*. Phys. Rev. Lett. (2003) **90**, page 068 702. ↗ http://www.phy.duke.edu/~socolar

[117] Sporns, Olaf; Chialvo, Dante R.; Kaiser, Marcus and Hilgetag, Claus C. *Organization, development and function of complex brain networks*. Trends in Cognitive Sciences (September 2004) **8** (9), pages 418–425. doi:10.1016/j.tics.2004.07.008

[118] Stauffer, Dietrich and Weisbuch, Gerard. *A Market of Inhomogeneous Threshold Cellular Automata*. Internat. J. of Modern Physics B (April 2003) **17** (29), pages 5495–5501. doi:10.1142/S0217979203023173

[119] Steuer, Ralf. *Computational approaches to the topology, stability and dynamics of metabolic networks*. Phytochemistry (2007) **68** (16–18), pages 2139–2151. doi:10.1016/j.phytochem.2007.04.041

[120] Steuer, Ralf; Gross, Thilo; Selbig, Joachim and Blasius, Bernd. *Structural kinetic modeling of metabolic networks*. Proc. Natl. Acad. Sci. USA (2006) **103** (32), pages 11 868–11 873. doi:10.1073/pnas.0600013103

[121] Tenenbaum, Gerald. *Introduction to Analytic and Probabilistic Number Theory* (Cambridge University Press, 1995), 2nd edition. ISBN 0521412617, 9780521412612

[122] Uchida, Satoshi. *Stability and Structure of Model Food-webs with Adaptive Behavior*. Ph.D. thesis, Technische Universität Darmstadt Dissertation D17 (2008). ↗ http://tuprints.ulb.tu-darmstadt.de/965/

[123] Watts, Duncan J. and Strogatz, Steven H. *Collective dynamics of 'small-world' networks*. Nature (June 1998) **393** (6684), pages 440–442. doi:10.1038/30918

[124] Weber, Sebastian; Hütt, Marc-Thorsten and Porto, Markus. *Pattern formation and efficiency of reaction-diffusion processes on complex networks*. Europhys. Lett. (2008) **82**, page 28 003. doi:10.1209/0295-5075/82/28003

[125] Weber, Sebastian and Porto, Markus. *Generation of arbitrarily two-point-correlated random networks*. Phys. Rev. E (October 2007) **76**, pages 046 111–9. doi:10.1103/PhysRevE.76.046111

[126] Wootters, William K. and Langton, Chris G. *Is there a sharp phase transition for deterministic cellular automata?* Physica D (September 1990) **45** (1–3), pages 95–104. doi:10.1016/0167-2789(90)90176-P

[127] http://wordnet.princeton.edu

[128] Zhang, J. and Small, M. *Complex Network from Pseudoperiodic Time Series: Topology versus Dynamics*. Phys. Rev. Lett. (June 2006) **96** (23), pages 238 701–238 704. doi:10.1103/PhysRevLett.96.238701

[129] Zhao, Yang and Doboli, A. *Finding broad-scale patterns in large size electronic circuit netlists*. In *Circuits and Systems, 2005. 48th Midwest Symposium on* (2005), volume 1 pages 307–310. doi:10.1109/MWSCAS.2005.1594100

Index

List of Figures

List of Tables

Curriculum vitae

Florian Greil (born 3 Aug. 1981 in Darmstadt)

Education

1990 – 2000	secondary school, Lichtenberg-Gymnasium Darmstadt final exam: Abitur
2000 – 2005	Major in physics at Technische Universität Darmstadt
May 2004 – Apr. 2005	Diploma thesis entitled *The influence of different updating schemes in random Boolean networks* Advisor: Prof. Barbara Drossel, Institut für Festkörperphysik `http://www.fkp.tu-darmstadt.de/drossel`
Jul. 2005	Graduation as *Diplom-Physiker*
Jul. – Sep. 2005	Scientific mission ARK-XXI/1a+b on board of the research ice breaker *Polarstern*, Alfred-Wegener-Institute for Polar and Marine Research Bremerhaven, Germany Task: Oceanographic measurements in the Fram Strait and the Greenland Sea `http://www.awi.de/infrastruktur/`
Nov. 2005 – May 2009	Working as research assistant on doctoral thesis
Sep. 2007 – Oct. 2008	Rearch visit in the Complex Systems Group in Calgary, Canada By invitation of Prof. Maya Paczuski and Prof. Jörn Davidsen `http://www.ucalgary.ca/complexity/`

Publications in refereed journals

1. Florian Greil, Kevin Bassler
   Attractor length distribution of critical Boolean networks
   In preparation.

2. Barbara Drossel, Florian Greil
   Critical Boolean networks with scale-free in-degree distribution
   Physical Review E **80**, 026102 (2009).
   `http://arxiv.org/abs/0901.0387`

3. Agnieszka Beszczynska-Möller, Rainer Graupner, Florian Greil, Kerstin Hans, Wolfgang Hayek, Ekkehard Schütt, Andreas Wisotzki, Matthias Monsees
   Flow through the Fram Strait and in the entrance to the Artic Ocean
   Berichte zur Polar- und Meeresforschung **570** (2008), ISSN 1618-3193.
   `http://epic.awi.de/Publications/Bud2008a.pdf`

4. Florian Greil, Barbara Drossel, Joost Sattler
   Critical Kauffman networks under deterministic asynchronous update
   New Journal of Physics, **9**, 373 (2007).
   http://arxiv.org/abs/0707.1450

5. Florian Greil, Barbara Drossel
   Kauffman networks with threshold functions
   European Physical Journal B **57** 109-113 (2007).
   http://arxiv.org/abs/cond-mat/0701176

6. Florian Greil, Barbara Drossel
   The dynamics of critical Kauffman networks under asynchronous stochastic update
   Physical Review Letters **95** 048701 (2005).
   http://arxiv.org/abs/cond-mat/0501081

7. Barbara Drossel, Tamara Mihaljev, Florian Greil
   Number and length of attractors in a critical Kauffman model with connectivity one
   Physical Review Letters **94** 088701 (2005).
   http://arxiv.org/abs/cond-mat/0410579

For simplicity, the hyperreference to the (freely accessible) pre-print version is given.

Acknowledgements

I would like to thank my supervisor Barbara Drossel for the opportunity to work in her group, for the chance to publish with her and for providing such a supportive team-based environment to work in. Without her support, this work would not have been possible.

Furthermore, I would like to thank all members of the working group, especially Agnes Szejka, Tamara Mihaljev and Tiago Paula de Peixoto for the numerous useful discussions during the group meetings.

Many thanks also to the complete complex system group headed by Maya Paczuski and Jörn Davidsen in Calgary, Canada for their invitation and the very productive atmosphere.

Gemäß der „Allgemeinen Bestimmungen der Promotionsordnung der Technischen Universität Darmstadt" vom 12. Januar 1990 muss eine Dissertation die nachfolgende Erklärung enthalten. Der Inhalt ist festgelegt in § 8 Abs. 1[a] und in § 9, Abs. 1 Satz 5[b].

Erklärung zur Dissertation

Hiermit erkläre ich eidesstattlich, dass ich die vorliegende Dissertation selbständig verfasst und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe. Ich erkläre ferner, dass ich bisher keinen Versuch unternommen habe, an einer anderen Hochschule das Promotionsverfahren einzuleiten.

[a] Der Absatz lautet in der Fassung der VI. Änderung vom 15. Februar 2006 wörtlich: „Das Promotionsverfahren wird eingeleitet durch ein schriftliches Gesuch des Doktoranden, das an den Dekan des zuständigen Fachbereiches zu richten ist. Dem Promotionsgesuch sind beizufügen: (a) eine Übersicht des Lebens- und Bildungsganges, (b) die Dissertation in Schriftform, in mindestens drei Ausfertigungen, (c) eine Erklärung, ob schon früher eine Promotion versucht wurde, gegebenenfalls mit näheren Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuches."

[b] Hier steht „Die Dissertation ist vom Bewerber mit einem Verzeichnis aller benutzten Quellen und einer Erklärung zu versehen, dass er die Arbeit – abgesehen von den in ihr ausdrücklich genannten Hilfen – selbstständig verfasst hat.".